AD-779 754

USER'S MANUAL FOR QUANTO-A WEAPON ALLOCATION CODE

Karl T. Benson, et al

Air Force Weapons Laboratory
Kirtland Air Force Base, New Mexico

April 1974

UNCLASSIFIED
Security Classification

AD779 754

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Air Force Weapons Laboratory (SAB) Kirtland Air Force Base, New Mexico 87117	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP
3. REPORT TITLE USER'S MANUAL FOR QUANTO - A WEAPON ALLOCATION CODE	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Final report; 1 September 1971-1 October 1973	
5. AUTHOR(S) (First name, middle initial, last name) Karl T. Benson, Capt, USAF; Craig E. Miller, Capt, USAF	

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
April 1974	130	None

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S) AFWL-TR-74-20
b. PROJECT NO. 8809	
c. Task No. 09	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	AFWL (SAB) Kirtland AFB, NM 87117

13. ABSTRACT

(Distribution Limitation Statement A)

This report provides instructions for using the QUANTO computer code, a code developed within the Analysis Division of the Air Force Weapons Laboratory (AFWL) to study the vulnerability of aircraft flushing from a nuclear attack from a force of sea-launched ballistic missiles (SLBMs). The structure of the input deck, the array dimensions of concern to the user, the special significance of selected input parameters, and the required job control language are described. The most current version of the QUANTO code is attached as Appendix I. Additional information concerning QUANTO is contained in the AFWL Technical Report AFWL-TR-73-242.

129

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
QUANTO						
Allocation						
Flush						
Input data						
Job control						
Optimization						
Survivability						
User's manual						
Utilization						

UNCLASSIFIED
Security Classification

AFWL-TR-74-20

USER'S MANUAL FOR QUANTO - A
WEAPON ALLOCATION CODE

Karl T. Benson
Captain USAF

Craig E. Miller
Captain USAF

Final Report for Period 1 September 1971 through 1 October 1973

TECHNICAL REPORT NO. AFWL-TR-74-20

Approved for public release; distribution unlimited.

FOREWORD

The research was performed under Program Element 62601F, Project 8809, Task 09.

Inclusive dates of research were 1 September 1971 through 1 October 1973. The report was submitted 18 January 1974 by the Air Force Weapons Laboratory Project Officer, Major Arthur R. Geldbach (SAB).

In the development of the user's manual for the QUANTO code, the experience and advice of Mr. William Peay and Mr. Eugene Omoda have been invaluable in deciding what information to include, in clarifying the user instructions, and in making the code easy to use efficiently.

This technical report has been reviewed and is approved.

Arthur R. Geldbach

ARTHUR R. GELDBACH
Major, USAF
Project Officer

GEORGE H. DIMON, JR.
Colonel, USAF
Chief, Battle Environments Branch

Charles C. Hyre

CHARLES C. HYRE, JR.
Colonel, USAF
Chief, Analysis Division

CONTENTS

TABLES

SECTION I

INTRODUCTION

The QUANTO computer code and its mathematical model were developed within the Analysis Division of the Air Force Weapons Laboratory (AFWL) to study the effects of a sea-launched ballistic missile (SLBM) nuclear attack on targets consisting of a flushing aircraft force. Using the technique of Lagrangian multiplier optimization, a near-optimal allocation of SLBMs to targets is produced. In addition, the submarine positions or the aircraft beddown are optimized if requested. The code has considerable flexibility in the type of input data it will permit through (1) its automatic consideration of specific values of aircraft hardnesses, flyout profiles, level-off altitudes, and kill values and (2) its treatment of an attack by multiple types of SLBMs, which are described by differing missile trajectories, yields, launch intervals, reliabilities, and numbers of missiles per submarine.

The report contains information which enables the user of QUANTO to construct a data deck and make a successful run. In addition, the limits on the numbers of data array elements currently permitted in a QUANTO run are described. However, the user is provided instructions for redimensioning arrays to accommodate larger problems or to reduce core requirements for smaller problems. The most recent version of the program is listed in appendix I. Sample control language and comments concerning several of the input parameters are also included.

This document is intended as a supplement to the AFWL Technical Report AFWL-TR-73-242, "QUANTO--A Code to Optimize Weapon Allocations," which describes the mathematical model and methods of optimization used by QUANTO. An example of an input deck, its output, and brief descriptions of the major subroutines appear in the referenced report.

SECTION II

USING THE PROGRAM

The user of QUANTO must construct a data deck and verify that the dimensions of arrays in DIMENSION and COMMON statements are sufficient for the problems described. The format of the input data will first be described. Then a discussion will be given of the DIMENSION and COMMON statements of concern when problem size forces enlargement or adjustment of array sizes.

1. STRUCTURE OF INPUT DECK

The input deck for a single problem is arranged in sections in the following order:

a. Initial data card

b. Beddown data

c. Aircraft parameters

d. Submarine positions

e. Missile data

f. Aircraft profile and nuclear effects parameters

g. Convergence parameters

h. Initial allocation

When multiple problems are to be run in a single job submission, the data decks for all problems may be stacked and input in a single stream.

a. Initial Data Card

The first card of any input problem contains basic data necessary to describe the problem. This card is described in table 1. Table 2 shows options.

b. Beddown Data

The aircraft beddown data is input as a set of cards for each base capable of having alert aircraft of any kind. The first card(s) of the set identifies the base. Each target (base) identification is followed by a take-off sequence list for alert aircraft on that base. The formats of these cards are described in tables 3 and 4.

Table 1

INITIAL DATA CARD FORMAT

(FORMAT (14I5))

Columns	Program variable name	Description
1-5	NTGTS	Number of bases in the list of aircraft beddown.
6-10	NSUBS	Number of candidate positions for submarines in the list of submarine positions.
11-15	NTYPES	Number of types of aircraft.
16-20	MXRWAY	Maximum number of runways on any one base.
21-25	MTYPES	Number of types of missiles.
26-30	IOUT	A control variable to limit output. If IOUT=2, intermediate multiplier and allocation output, useful for debugging, will be output during the convergence to the optimal laydowns. If IOUT=1, the intermediate output mentioned above will be suppressed.
31-35	ISOPT	A control variable to control whether or not optimization of submarine positioning among the candidate positions is performed. If ISOPT=1, submarine positioning will NOT be optimized among the input submarine locations. If ISOPT=2, submarine positioning will be optimized.
36-40	IVOPT	A control variable to control whether or not the beddown of aircraft will be optimized among the given bases. If IVOPT=1, the beddown will NOT be optimized. If IVOPT=2, the beddown will be optimized. NOTE: If both ISOPT and IVOPT are set equal to 2, the submarine positioning is optimized first, then fixed in these locations for the beddown optimization.
41-45	NCASE	An arbitrary "case number" which the user may use to identify his problem.
46-50	MODE	A multipurpose control variable which is used to specify one of several operational modes. Table 2 describes the mode options.

Table 2

MODE OPTIONS

Mode	Purpose	Input	Execution and termination
0	The user may edit the major portion of his input to see if it is read properly by QUANTO without risking the computer time required for optimizations. At the same time, the QUANTO tasks of constructing the aircraft profiles and computing lethal areas may be checked for validity and exceptional conditions which may not have been foreseen.	Problem description data, without the convergence parameters and initial allocation, is read from data cards.	Execution of problem terminates after processing of aircraft profile and nuclear effects information. No restart tape is written.
1	The reading of the entire input deck and the computations of survivabilities and kill from the initial laydown may be tested without risking the computer time required for the iterative optimizations. If all operations are satisfactory, the job may be continued using the results of these computations by reading the created restart tape with a MODE=2 job.	Complete input deck is read from data cards.	Execution terminates after computation of survivabilities and expected kill from initial allocation. A restart tape (with all necessary data for continuded operation with MODE=2) is written.
2	A MODE=1, 2, or 3 job may be continued using the results of all computations completed when the previous job terminated. A restart tape from the previous job supplies the intermediate results.	Intermediate problem description data is read from restart tape. Only initial data card is input in card format.	Execution terminates following the writing of a restart tape after any laydown convergence which terminates within 30 seconds of the job's time limit.

Table 2 (cont'd)

Mode	Purpose	Input	Execution and termination
3	The entire problem, from input of the data deck through optimization, may be attempted. Computer time may be wasted in the optimization if input errors exist. If the time limit is reached, all intermediate computational results will be preserved on a restart tape.	Complete input deck is read from data cards.	Execution terminates as for MODE=2.

Table 3*

TARGET IDENTIFICATION CARD(S) FORMAT

(FORMAT (3F10.4, I5, 5X, 3F10.4/(7F10.4)))

Columns	Program array name	Description
1-10	TGTLAT	Latitude coordinates of the base's brake release point (start of take-off roll), in degrees (northern hemisphere is assumed).
11-20	TGTLNG	Longitude coordinates of the base's brake release point (start of take-off roll), in degrees (west of Greenwich mean line is assumed).
21-30	DTCENT	Distance from brake release point to the centroid of the dispersing aircraft, in nautical miles (NM).
31-35	NRWAYS	Number of runways, used to determine take-off intervals between aircraft on this base.
41-50 51-60 61-70 Following card(s) if necessary in 7F10.4 format.	VAL	Number of aircraft of Type 1, then Types 2, 3, ... , until the numbers of all types of aircraft on this base have been listed.

*Note that if only three or fewer types of aircraft are included in the problem and no base has more than 70 alert aircraft, then only two cards will be necessary for each base, so the beddown data will consist of consecutive pairs of cards, one pair per target.

Table 4*

TAKE-OFF SEQUENCE CARD(S) FORMAT

(FORMAT (70I1))

Columns	Program array name	Description
1,2,3,... 70 1,2,...	ISEQ	The type numbers of the alert aircraft on the base, in the order of take-off, for as many cards as needed.

*Note that if only three or fewer types of aircraft are included in the problem and no base has more than 70 alert aircraft, then only two cards will be necessary for each base, so the beddown data will consist of consecutive pairs of cards, one pair per target.

c. Aircraft Parameters

The point values assigned to each aircraft, brake release times (for each type of aircraft if first to take off), aircraft hardnesses, and take-off intervals follow the beddown data in the input deck. First, one card per aircraft is input, in order by aircraft type number, giving the first three of these items; this card will be called the "aircraft card." Then the take-off intervals are listed in a specific order on the subsequent card(s). The formats of these cards are described in tables 5 and 6.

Table 5

AIRCRAFT CARD FORMAT

(FORMAT (3F10.4, I5))

Columns	Program array name	Description
1-10	RELVAL	The number of value points each aircraft of the given type is worth to the enemy when killed.
11-20	BRTIME	The time between launch of the first SLBMs and the brake release time (start of take-off roll) of the given type aircraft, if that type aircraft is the first to take off at any base, in minutes.
21-30	PSI	The number of psi (pounds per square inch) of over-pressure which kills the aircraft.
31-35	ICAL	The number of cal/cm^2 of incident free-field thermal energy which kills the aircraft.

Table 6

TAKE-OFF INTERVALS CARD FORMAT

(FORMAT (7F10.4))

Columns	Program array name	Description
1-10 11-20 21-30 etc., through 70; then repeat on subsequent cards.	DELTAC	The take-off intervals between each ordered pair of types of aircraft, for bases with each possible number of runways, are listed in specific order. First, for single runways, the intervals between the aircraft-type pairs (1,1),(1,2),(1,3),...,(1,NTYPES),(2,1),(2,2),...,(2,NTYPES),(3,1),...,(NTYPES,NTYPES), are input in that order. A set of intervals in this order is then input for dual runways, then triple, etc., until all numbers of runways which occur on input bases have been covered.

d. Submarine Positions

One data card is input per candidate submarine location. A submarine location is characterized not only by the type of missile (or submarine) which may be located there, but by the number of missiles on that type of submarine. Type numbers are applied to submarine locations to indicate the type of submarine which may be located there. In the automated relocation of submarines, shifts of submarines may occur only between locations which may have like types of submarines. So the numbering of submarine types may be constructed so as to prevent submarines from moving to certain points or jumping from one ocean to another. If more than one type of submarine can be at the same location, that point must be input once for each candidate type of submarine. The format of the submarine location card is described in table 7.

Table 7

SUBMARINE LOCATION CARD FORMAT

(FORMAT (2F10.4, 3I5))

Columns	Program array name	Description
1-10	SUBLAT	Latitude coordinates of the candidate submarine location in degrees (north hemisphere is assumed).
11-20	SUBLNG	Longitude coordinates of the candidate submarine location in degrees (west of Greenwich mean line is assumed).
21-25	ISUBS	Number of submarines stationed at the submarine location, prior to submarine relocation (if requested). This may be zero.
26-30	NMPS	Number of missiles on all submarines at the location which will be dedicated to aircraft kills.
31-35	MTYPE	Type-number of the submarines (or SLBMs) which may be located at the point.

e. Missile Data

The type numbers appearing in the submarine location input refer to the type of submarine. Since each submarine can carry, at most, one type of missile, the submarine type is equivalent to the missile type. For each type submarine, parameters describing the SLBM must be input. Sets of cards describing each

SLBM are input sequentially by submarine type number. Each set of cards for a single SLBM contains

(1) The SLBM parameters card

(2) The missile trajectory card(s)

If two different submarine type numbers are used for the same type SLBM merely to prevent submarines from shifting to certain points, the set of cards describing that SLBM must be input twice. The formats of these cards are described in tables 8 and 9.

Table 8

SLBM PARAMETERS CARD FORMAT
(FORMAT (7F10.4))

Columns	Program array name	Description
1-10	DELTM	The time interval between salvo launches for successive missiles of the given type, in minutes.
11-20	RELML	The launch reliability of the missile (i.e., the percent which launch successfully on the average).
21-30	RELMF	The in-flight reliability of the missile (i.e., the percent which successfully reach the target area on the average).
31-40	RELMWH	The warhead reliability of the missile (i.e., the percent which successfully detonate upon reaching the target on the average).
41-50	RNGMIN	The minimum range of the missile, in NM.
51-60	RNGMAX	The maximum range of the missile, in NM.
61-70	YIELD	The assumed yield of the missile warhead, in KT.

The tabular time/distance pairs of table 9 are used to compute missile arrival times on each target, with four-point Lagrangian interpolation supplying times for distances not appearing in the table.

Table 9

MISSILE TRAJECTORY CARD(S) FORMAT
(FORMAT (I5, 5X, (6F10.4)))

Columns	Program variable or array name	Description
1-5	MPR	Number of time/distance pairs which describe the times associated with the missile trajectories to various distances. These pairs immediately follow MPR.
11-20	FMTIME	The flight time, in minutes, which the missile requires to reach a target a given number of NM from the launch point, where this distance is given immediately following this time on the input card.
21-30	FMRNG	The distance, in NM, corresponding to the preceding missile flight time.
31-40	FMTIME	Succeeding time/distance pairs, in ascending order by time, which describe the missile flight times.
41-50	FMRNG	
51-60	FMTIME	
61-70	FMRNG	

Succeeding cards in (6F10.4) format until all MPR pairs have been input.

f. Aircraft Profile and Nuclear Effects Parameters

A set of cards is input for each aircraft type, with the sets in order by aircraft-type number. Each set consists of the following:

(1) Altitudes card

(2) Count card

(3) Aircraft flight profile data cards

(4) Control variables card

(5) Nuclear effects parameters card

(6) Lethal radii cards

The formats of these cards are described in the following tables.

(1) Altitudes Card

This card indicates the final level-off altitude of the aircraft of the given type and the height of burst (HOB) of the SLBMs. Both of these altitudes are relative to the ground. Thus, if each base has only one type of

aircraft stationed there, different HOBs may be selected for each type aircraft. However, this will cause misleading results if a base has more than one type of aircraft, since the lethal area computations for one SLBM against two different types of aircraft will be based on two different HOBs for the single weapon. The format of this card appears in table 10.

Table 10

ALTITUDES CARD FORMAT

(FORMAT (6F10.4))

Columns	Program variable name	Description
1-10	AZ	The final level-off altitude of the aircraft in its flyout profile, in feet above the ground.
11-20	HBL	The detonation height of SLBMs used in computing lethal areas for the aircraft, in feet above the ground.

(2) Count Card

The count card indicates only the number of cards following it which contain the aircraft profile information. Columns 1 through 5 of this card should contain this integral number of cards, and the remaining columns may be used for comments.

(3) Aircraft Flight Profile Data

The aircraft flight profile data is input in a set of data cards which describe the aircraft movement in time and space. The flight profile for a given type of aircraft consists of values for distance, time, altitude, and the horizontal components of velocity and level-off acceleration (as a function of altitude). These values are specified for a nonturning aircraft.

Distance and time values are measured relative to brake release, the point at which the aircraft begins moving on the runway for take-off. The values for altitude are measured relative to ground level and must be non-decreasing. The level-off altitude may be specified as any altitude less than or equal to the maximum (i.e., las) altitude value of the aircraft flight profile. The values for velocity, distance, time, and acceleration of the aircraft when it levels off are obtained through interpolation among the

altitude values of the profile based on the input level-off altitude. The parameters associated with level-off are used in the standard equations of motion to accelerate the aircraft to its final escape velocity. The final escape velocity is assumed to be the maximum velocity, expressed as Mach number, which occurs in the input aircraft flight profile data, although it is input separately as FMACH in the control variables card to be described next. An exception to this rule is made when XATI=0.0 (in the control variables card--see next paragraph), in which no acceleration takes place after level-off. The user is required to input values for altitude which are nondecreasing, and these values are automatically adjusted, if necessary, to ensure strict monotonicity for purposes of interpolation. The aircraft flight profile description should consist of approximately the maximum of 50 cards currently permitted by array dimensions in QUANTO. The format and content of each card are described in table 11.

Table 11

AIRCRAFT FLIGHT PROFILE DATA CARD FORMAT
(FORMAT (3F15.8, 2F10.6))

Columns	Program array name	Description
1-15	F	The ground range from the brake release point, in feet, indicating the distance traveled in the following time.
16-30	G	The time from brake release, in seconds, at which the aircraft reaches the preceding distance.
31-45	A	The altitude of the aircraft, in feet above ground elevation, at the preceding time.
46-55	VEL	The Mach number describing the aircraft's ground speed at the time indicated in this card.
56-65	ACCEL	The acceleration, in feet per second per second at which the aircraft would accelerate to its final escape velocity if it leveled off at a lesser velocity at the altitude on this card.

(4) Control Variables Card

This card supplies the values of variables which control the operation of generating a completed aircraft flight profile for use in the

program. The format of this card is described in table 12, and additional information concerning the variables is in the following narrative.

The variables, FALTCM and FMACH, are determined by the input aircraft flight profile. The aircraft is assumed to follow the input flight profile until the level-off altitude is reached. If XATI>0.0, the aircraft is then accelerated to its final escape velocity which is assumed to be the maximum Mach number which occurs in the input profile and is specified separately as FMACH. The lowest altitude in the input profile data at which the aircraft reaches this final escape velocity is specified by FALTCM. If the level-off altitude is less than FALTCM, then the velocity upon leveling off is less than FMACH. The standard equations of motion are then used to accelerate the aircraft to its final escape velocity, if the variable XATI is greater than zero.

The variable XATI specifies the number of points which QUANTO generates in accelerating the aircraft to its final escape velocity. If XATI is equal to 0.0, then the velocity upon leveling off is used as the final escape velocity, and no acceleration takes place. The variable XATI should be chosen such that a sufficient number of points of distance and time are generated for the acceleration phase to permit reasonably accurate interpolation.

The distance associated with the point that the aircraft achieves its final escape velocity is usually not great enough for the lethal area determination routines to use in computing lethal area of the most distant SLBM weapon detonations. The variable TI(I) is used to extend the profile to the greatest distance needed. This is the greatest distance from the centroid that is necessary to compute lethal area for the placement of the last potential weapon arriving on that target.

The choice of TI(I) and XATI must be made so that the complete aircraft profile is generated with sufficient spacing in the data to accommodate four-point Lagrangian (cubic) interpolation. An upper limit of 99 total data points in the complete aircraft flight profile is permitted by the present array dimensions in QUANTO. Currently used values for TI(I) and XATI are 60 seconds and 10 points, respectively.

Table 12

CONTROL VARIABLES CARD FORMAT
(FORMAT (3F15.8, 2F10.6))

Columns	Program variable name	Description
1-15	FALTCM	The altitude, in feet above ground elevation, from the aircraft flight profile, at which the aircraft first reaches its final escape velocity.
16-30	FMACH	The final escape velocity of the aircraft, expressed as Mach number, from the aircraft flight profile. The aircraft will not exceed this velocity during flush.
31-45	TI(I)	The time interval, in seconds, at which data points for distance will be generated after the aircraft (of type I) reaches its final escape velocity.
46-55	XATI	The number of equally spaced points in time for which distances will be generated to accelerate the aircraft from its velocity at its level-off altitude to its final escape velocity.

(5) Nuclear Effects Parameters Card

This card contains parameters which indicate the environment in which the nuclear detonations take place. These parameters are arguments required for the nuclear effects subroutines SABERCM and SNAPTCM. Table 13 describes the format of this card.

(6) Lethal Radii Cards

If the lethal overpressure radius and time of shock arrival and/or the lethal thermal radius are already known for the hardnesses of some or all aircraft types against some or all missile types for the case in which the aircraft has reached its terminal altitude by the time it encounters the lethal region, some of the computations of SABERCM and SNAPTCM may be bypassed by inputing the necessary values in cards following the nuclear effects parameters card. A pair of cards, the "overpressure card" and the "thermal card," must be input for each combination of aircraft type and missile type. These pairs are input in ascending order by aircraft type and by missile type within aircraft type. These cards must be input even if the values are not known, but the cards may be left blank. The card formats are in tables 14 and 15.

Table 13

NUCLEAR EFFECTS PARAMETERS CARD FORMAT

(FORMAT (6F10.8))

Columns	Program variable name	Description
1-10	HTE	The assumed ground elevation, in feet above sea level, of the target air bases from which this aircraft flies. This currently must be zero since QUANTO's computations of lethal areas are not reaccomplished for each base.
11-20	BETIND	The "beta ID," an indicator which the subroutine SNAPTCM requires to compute the effect of thermal energy on the aircraft by considering the aircraft as a horizontal panel (when BETIND=0.0) or as a panel oriented for maximum perpendicular incident thermal energy (when BETIND=1.0). QUANTO uses BETIND=1.0, assuming some panel of the aircraft is in the worst orientation.
21-30	RHO	The albedo factor, which is the fraction of incident thermal energy which reflects off the ground. For a SAC Normal Day, the albedo is 0.3. A worst case abnormal day has a higher albedo.
31-40	VIS	The visibility in miles; 10 for a SAC Normal Day.
41-50	PZ	The water vapor in the air, in millimeters; 5 for a SAC Normal Day.
51-60	HSL	The altitude in feet above sea level of the top of the haze layer; 10,000 for a SAC Normal Day.

Table 14

OVERPRESSURE CARD FORMAT

(FORMAT (I5, 2F15.8))

Columns	Program variable name	Description
1-5	ISABER	An indicator of whether the following two fields of this card are to be used. If ISABER=1, the values are to be used; otherwise, input ISABER=0 and QUANTO will compute the values.
6-20	HORF	The lethal overpressure radius, in feet.
21-35	TSA	The time of shock arrival, in seconds.

Table 15

THERMAL CARD FORMAT

(FORMAT (I5, 2F15.8))

Columns	Program variable name	Description
1-5	ISNAPT	An indicator of whether the following field on this card is to be used. If ISNAPT=1, the value is to be used; otherwise, input ISNAPT=0, and QUANTO will compute the value.
6-20	SZ	The lethal thermal radius, in feet.

g. Convergence Parameters

Several parameters may be input to control the termination of QUANTO's iterative procedure for converging the λ_{ij}'s in computing the optimal laydown. As described in AFWL-TR-73-242, the procedure terminates (1) when the λ_{ij} matrix is converged to some tolerance ε, (2) when the expected kill does not significantly increase for a given number of iterations, or (3) when the number of iterations reaches a maximum specified by the user. The convergence parameters are input by the user in the format described in table 16 to indicate the final tolerance ε, the number of kills used for testing for a "significant increase," the interval of iterations over which the kill increase is measured, and the maximum number of iterations. Current values of the parameters listed in table 16 are 0.01, 20, 100, and 0.0001, in the order of the table.

h. Initial Allocation

A missile laydown must be input to QUANTO to start the iterative procedure for improving the laydown. The initial allocation of missiles to targets is input submarine by submarine. The submarines are in the order of the submarine locations, with all submarines at one location preceding the first submarine at the next location. One or more cards, in the format (14I5), are input for each submarine where the numbers on the cards are the target numbers to which the SLBMs from that submarine (in ascending order by salvo or launch time) are allocated. One card per submarine is sufficient if each submarine has 14 or fewer missiles; otherwise, multiple cards per submarine are needed with a new card started for each new submarine.

Table 16

CONVERGENCE PARAMETERS FORMAT

(FORMAT (F10.4, 2I5, F10.4))

Columns	Program variable name	Description
1-10	CHGKIL	The number of kills used to distinguish between a significant kill increase (in ITCUT2 iterations--see below) and an insignificant increase.
11-15	ITCUT1	The maximum number of groups of ITCUT2 iterations (see below) which will be accomplished by QUANTO in any laydown optimization procedure before the iteration procedure is terminated by QUANTO.
16-20	ITCUT2	The number of iterations in each group, where the increase in expected kill resulting from each group of iterations is tested against the amount CHGKIL to determine if the increase was large enough to warrant continuing the iterations through another group.
21-30	EPSCUT	A tolerance quantity used to test for convergence of the λ_{ij} matrix. QUANTO actually converges the λ_{ij} matrix to tolerances of ε=.1, then 0.01, then 0.001, etc., until convergence to a tolerance $\varepsilon \leq 1.5$ times EPSCUT is achieved.

2. SPECIFICATION OF ARRAY DIMENSIONS

Since array dimensions as currently set in QUANTO may not be sufficient for some types of problems, the arrays which have dimensions of interest to the user are listed in table 17 with the problem-dependent dimensions indicated as program variable names described in table 13. Only two of these program variables, MXTGT and MXWPNS, actually appear in the QUANTO code and must be set in the QUANTO main program (in a DATA statement). The others cross-reference the two tables and represent constants which may need to be changed (in the routines indicated) for problems which require more capacity than the constants of table 18 permit. In the execution of QUANTO, the maximum index of elements accessed in the arrays is controlled by program variables, so the constants specified in the DIMENSION and COMMON statements must be at least as large as (but may be larger than) the controlling program variables input through (or computed from inputs in) the data deck. Complete descriptions of those variables directly input may be found in the description of the input deck.

To conserve core, the user may want to reduce the array dimensions to the minimum required to run his problems. Furthermore, if only one type of aircraft occurs in his problems, the arrays FLAMB and FLAMBI may be equivalenced in the main program QUANTO.

3. SAMPLE QUANTO RUN CONSIDERATIONS

The listing of QUANTO in appendix I is the most current one as of the time this report was produced. For convenience, references to the lines of the program will use the alphanumeric identifiers on the right side of each line. Thus, the first line of the program is line QUA 10. This section will present the reader with the required considerations for several typical problems.

a. Considerations Concerning the Number of Aircraft Types

As the program is listed in appendix I, up to 35 bases, 168 weapon groups, one type of aircraft, etc. (in accordance with the figures of table 18) are permitted in the problem(s) to be run. Note that only <u>one</u> aircraft type is permitted since the arrays FLAMB and FLAMBI are equivalenced in line QUA 160. In view of this fact, core requirements could be reduced (when the problem has only one aircraft type) by changing the dimensions in lines QUA 30, QUA 50, QUA 70, and QUA 110 to the following, respectively:

```
1 SURV(35,168,1),FLAMBI(35,168,1),
3 VAL(35,1),PROD(35,1),VKILL(35,1),
5 RELVAL(1),BRTIME(1),PSI(1),ICAL(1),PKCIR(1),PKAN(1),
9 QPTS(30),QAREAL(30,1,4),QRLMX(30,1,4),DELTAC(2,1,1),JFLAC(2,1)
```

The user may quickly determine all of the occurrences of dimensions permitting two types of aircraft by searching table 17 for all occurrences of the dimension MXNTYP. In the QUANTO version in appendix I, MXNTYP has the value 2 wherever it occurs. Besides the above four lines, MXNTYP also occurs in the COMMON's labeled DISTIME, ACFTS, and NUCLER. Unfortunately, changing the dimensions in these labeled COMMON statements requires changing several lines since the COMMON statements appear in multiple routines, as listed in table 17. The user will probably want to weigh the inconveniences of changing QUANTO program statements against the core savings to be gained. For instance, the just described changes of lines QUA 30, QUA 50, QUA 70, and QUA 110 account for core savings of 11760, 105, 6, and 247 locations, respectively. The dimensions which appear in the QUANTO version in appendix I are intended to be adequate for most problems. If all these dimensions are adequate for the user's

problems, but the problem has two types of aircraft, the user need only delete the EQUIVALENCE statement (line QUA 160) for QUANTO to run the problem.

b. Changing the Numbers of Targets and/or Weapon Groups

By studying the first 16 lines of the QUANTO listing in appendix I, the user may determine that core requirements are most critically dependent on the numbers of targets, weapon groups, and types of aircraft. In particular, changes in lines QUA 20, QUA 30, QUA 150, and QUA 160 will usually bring the greatest payoff in core savings when the problems to be run have fewer than 35 targets, 168 weapon groups, and/or two types of aircraft. The considerations concerning the numbers of types of aircraft have already been discussed in paragraph a above.

When the problem has fewer than 35 targets, core savings will result from substituting the number of targets for 35 wherever 35 occurs in lines QUA 20, QUA 30, and QUA 150. In addition, if more than one type of aircraft is used, then the dimensions of the arrays VAL, PROD, and VKILL in line QUA 50 must be made consistent with the dimension changes in lines QUA 20, QUA 30, and QUA 150. Of course, the dimension MXTGT occurs elsewhere, as may be seen in table 17, but relatively small core savings result from changes elsewhere.

Likewise, the number of weapon groups (which is the sum of salvos from separately input submarine locations) may be reduced from 168 by changes in lines QUA 20, QUA 30, and QUA 150. These are all the lines in which the dimension MXWPNS (of table 18) occurs.

Changes in the dimensions must be consistent with other changes in the input deck. Dimensions must be at least as large as required by the input, or results will be unpredictable. A change in the number of targets will be reflected in the initial data card and the beddown data. A change in the number of weapon groups will be reflected in the initial data card, the number of submarine positions or salvos from those positions, and the initial allocation of SLBMs.

c. Output Options

The variable IOUT input in the initial data card indicates when an expanded output, sometimes useful for debugging, is desired. This output should never be requested for production runs and is only occasionally useful to the programmer analyzing program operations. Consequently, IOUT=1 should

always be input for production runs. Even then, voluminous output will be produced if beddown optimization is requested (i.e., IVOPT=2), and considerable output is produced by many other types of problems. Therefore, microfilm output is recommended until the volume of output is well known.

d. Distance to Centroid Computations

A side computation is required before the distances to the centroids may be estimated and input to QUANTO. The centroids of the areas of dispersing aircraft are dependent on the flyout profile and the geometry of turning. A rough approximation of the centroid location, for aircraft departing a single runway, may be obtained by computing (1) the location of an aircraft which turns 180 degrees following take-off at the time at which the aircraft reaches its terminal velocity, and (2) the location of that same aircraft after the same period of time, assuming it had not turned. The midpoint of these two locations would then be the approximate location of the centroid. A program is under development which will compute the centroid location giving equal weight to each possible aircraft location for turns of from 0 to 180 degrees in either direction. As the user might guess, considerable information about the flight characteristics of the aircraft is necessary for this computation. If multiple types of aircraft are used, the user must select one type as the one determining the centroid location.

Table 17

ARRAYS OF QUANTO WITH PROBLEM-DEPENDENT DIMENSIONS

Array name and dimensions	Array contents (and routines which have constant dimensions for array in COMMON or DIMENSION statements)
A. Arrays not in COMMON	
1. ALOC (MXTGT, MXWPNS)	The n_{ij} allocation matrix (in routine QUANTO).
2. FLAMB (MXTGT, MXWPNS)	The λ_{ij} matrix, whose definition depends on the number of aircraft types (in QUANTO).
3. SURV (MXTGT, MXWPNS, MXNTYP)	The S_{ij} survivability matrix for all aircraft types (in QUANTO).
4. FLAMBI (MXTGT, MXWPNS, MXNTYP)	The λ_{ij} matrices for all aircraft types (in QUANTO).
5. TGTLAT (MXTGT)	The latitudes of the targets (in QUANTO).
6. TGTLNG (MXTGT)	The longitudes of the targets (in QUANTO).
7. DTCENT (MXTGT)	The distances to the centroids of the areas of dispersed aircraft leaving the targets (in QUANTO).
8. NRWAYS (MXTGT)	The number of runways at each target (in QUANTO).
9. NWALOC (MXTGT)	The number of weapons allocated to each target (in QUANTO).
10. VAL (MXTGT, MXNTYP)	The value of the aircraft of each type on each target (in QUANTO).
11. PROD (MXTGT, MXNTYP)	The survivability products $\prod_{j=1}^{NWPNS} \left(S_{ij}^{\,n_{ij}}\right)$ for each target and each type aircraft (in QUANTO).
12. VKILL (MXTGT, MXNTYP)	The value of the aircraft of each type killed for each target (in QUANTO).
13. ISEQ (MXTGT, MXAC)	The take-off sequence on each base, as indicated by an ordered list of aircraft type numbers (in QUANTO).

Table 17 (cont'd)

Array name and dimensions	Array contents (and routines which have constant dimensions for array in COMMON or DIMENSION statements)
14. BRT (MXAC)	A working area used to compute the brake release times for each aircraft on a base (in QUANTO).
15. RELVAL (MXNTYP)	The point value assigned to a single aircraft of each type (in QUANTO).
16. BRTIME (MXNTYP)	The brake release times for each type aircraft, if that type takes off first from any base (in QUANTO).
17. PSI (MXNTYP)	The hardness of each aircraft type to overpressure; used to distinguish between kill and nonkill (in QUANTO).
18. ICAL (MXNTYP)	The hardness of each aircraft type to incident thermal energy; used to distinguish between kill and nonkill (in QUANTO).
19. PKCIR (MXNTYP)	A working area used to store the computed P_k for each aircraft type when the aircraft are assumed to be uniformly distributed within a circle (in QUANTO).
20. PKAN (MXNTYP)	A working area used to store the computed P_k for each aircraft type when the aircraft are assumed to be uniformly distributed within an annulus (in QUANTO).
21. SUBLAT (MXSUB)	The latitudes of the submarine locations (in QUANTO).
22. SUBLNG (MXSUB)	The longitudes of the submarine locations (in QUANTO).
23. ISUBS (MXSUB)	The number of submarines at each submarine location (in QUANTO).
24. NMPS (MXSUB)	The number of missiles on each submarine at each submarine location (in QUANTO).
25. MTYPE (MXSUB)	The type of missile on each submarine at each submarine location (in QUANTO).
26. ITGTNO (MXM)	A working area used to store the target numbers to which the missiles from a single submarine are initially allocated (in QUANTO).

Table 17 (cont'd)

Array name and dimensions	Array contents (and routines which have constant dimensions for array in COMMON or DIMENSION statements)
27. DELTM (MXMTYP)	The launch interval of each missile type (in QUANTO).
28. RELML (MXMTYP)	The launch reliability of each missile type (in QUANTO).
29. RELMF (MXMTYP)	The in-flight reliability of each missile type (in QUANTO).
30. RELMWH (MXMTYP)	The warhead reliability of each missile type (in QUANTO).
31. RNGMAX (MXMTYP)	The maximum range of each missile type (in QUANTO).
32. RNGMIN (MXMTYP)	The minimum range of each missile type (in QUANTO).
33. YIELD (MXMTYP)	The yield of each missile type (in QUANTO).
34. FMTIME (MXTRAJ, MXMTYP)	The times that are input in describing the trajectories of each type missile (in QUANTO).
35. FMRNG (MXTRAJ, MXMTYP)	The ground ranges that are input in describing the trajectories of each type missile (in QUANTO).
36. MPROF (MXMTYP)	The number of time/distance pairs describing the trajectories of each missile type (in QUANTO).
37. QAREAL (30, MXNTYP, MXMTYP)	A table of lethal areas for 30 distances of the detonation from the centroid for each aircraft and missile combination (in QUANTO).
38. QRLMX (30, MXNTYP, MXMTYP)	A table of distances from detonation point, describing the farthest reach of the lethal area in a direction away from the centroid, for 30 distances of the detonation from the centroid for each aircraft and missile combination (in QUANTO).
39. DELTAC (MXRWAY, MXNTYP, MXNTYP)	The take-off intervals between each pair of aircraft types from bases with each possible number of runways (in QUANTO).
40. JFLAC (2, MXNTYP)	A working area used to store pointers to the first and last aircraft (of each type) departing a base (in QUANTO).
41. ALHOLD (MXTGT)	A working area used to integerize the weapon laydown (in ALINT).

Table 17 (cont'd)

Array name and dimensions	Array contents (and routines which have constant dimensions for array in COMMON or DIMENSION statements)
42. VHOLD (MXTGT)	A working area used to integerize the beddown (in VINT).
43. SUMAC (5)	A working area used to compute the number of aircraft of each type on a single target for up to 5 aircraft types (in TGTKIL).
44. SUMACK (5)	A working area used to compute the number of aircraft killed of each type on a single target for up to 5 aircraft types (in TGTKIL).
45. LOHOLD (16) MXHOLD (16) LOTEMP (16) MXTEMP (16) LOH (16) MXH (16)	Working areas used in relocating submarines and assigning their SLBMs to new targets, for up to 16 missiles on a submarine (in SUBADJ).
B. <u>In COMMON named DISTIME</u> (arrays appear in routines QUANTO, PROCESS, DETAREA, BAKUP, TIMERAD, DATAGEN, and TIMEGEN with constant dimensions)	
1. FOTIME (MXPROF, MXNTYP) (also called S)	The times that are input and/or generated in describing the flight profile of each type aircraft.
2. FORNG (MXPROF, MXNTYP) (also called D)	The ground ranges that are input and/or generated in describing the flight profile of each type aircraft.
3. NPROF (MXNTYP) (also called NUMDATA)	The number of points of time and distance input and/or generated for the flight profile of each type aircraft.
4. CV (MXNTYP)	The final velocity reached by each aircraft type.
5. TI (MXNTYP)	The time interval used in generating profile points for each aircraft type.
C. <u>In COMMON named ACFTS</u> (arrays appear in routines PROCESS and DETAREA with constant dimensions)	
1. A (MXPIN, MXNTYP)	The altitude values input for each aircraft profile.
2. F (MXPIN, MXNTYP)	The ground range values input for each aircraft profile.
3. G (MXPIN, MXNTYP)	The time values input for each aircraft profile.

Table 17 (cont'd)

Array name and dimensions	Array contents (and routines which have constant dimensions for array in COMMON or DIMENSION statements)
4. VS (MXPIN, MXNTYP)	The velocities of sound for altitude values input for each aircraft profile.
5. VEL (MXPIN, MXNTYP)	The velocities input for each aircraft profile.
6. ACCEL (MXPIN, MXNTYP)	The level-off accelerations input for each aircraft profile.
7. NDATA (MXNTYP)	The number of input sets describing each aircraft profile.
D. In COMMON named NUCLER (arrays appear in routines PROCESS and DETAREA with constant dimensions)	
1. FLRP (MXNTYP, MXMTYP)	The computed lethal radii for overpressure for each missile type against each aircraft type.
2. FTSA (MXNTYP, MXMTYP)	The computed times of overpressure shock arrival for each missile type against each aircraft type.
3. FLRT (MXNTYP, MXMTYP)	The computed lethal thermal radii for each missile type against each aircraft type.
4. BURST (MXNTYP)	The heights of burst for each type aircraft.
5. DISMIN (MXNTYP)	The computed horizontal ground distances traveled by each type aircraft before it reaches its terminal altitude.
6. VPSI (MXNTYP)	The overpressure hardness of each type aircraft.
7. VCAL (MXNTYP)	The thermal hardness of each type aircraft.
8. VYIELD (MXMTYP)	The yield of each type missile.

Table 18

PROBLEM-DEPENDENT DIMENSIONS IN QUANTO

Variable dimension	Description	Current QUANTO dimension
MXTGT	The maximum number of targets.	35
MXWPNS	The maximum number of weapon groups. A weapon group includes all potential missiles of the same type launched from the same point at the same time. The total number of weapon groups in a given problem may be computed easily by summing the numbers of missiles per submarine (NMPS of table 7) over all separately input potential submarine locations.	168
MXNTYP	The maximum number of types of aircraft.	2*
MXAC	The maximum number of aircraft on any base.	34
MXRWAY	The maximum number of runways at a base.	2
MXSUB	The maximum number of submarine locations.	28
MXM	The maximum number of missiles on a single submarine.	16
MXMTYP	The maximum number of missile types.	4
MXTRAJ	The maximum number of time/distance pairs input in describing missile trajectories.	15
MXPROF	The maximum number of time/distance pairs input and/or generated in describing aircraft flyout profiles.	99
MXPIN	The maximum number of input sets describing an aircraft profile.	50

*Although two types of aircraft are permitted by the dimension statements in QUANTO, the "EQUIVALENCE (FLAMB, FLAMBI)" statement in QUANTO must be removed before running a problem having two types of aircraft.

SECTION III

TIPS ON THE UTILIZATION OF QUANTO

This section describes the flow of information to and from the program QUANTO and presents the user with the job control language (in SCOPE Version 3.2) required to execute QUANTO on the CDC 6600 computer system at the Air Force Weapons Laboratory (AFWL), Kirtland AFB, New Mexico. No attempt is made to describe the job control language functions in detail, but the reader who is familiar with some computer operating system should be able to generate similar operations by referring to his own job control language manuals and the discussion and examples presented here.

1. INFORMATION FLOW

The term QUANTO has been used to refer to the aggregation of all routines required to make optimization runs. The user deals with the program (making modifications, changing dimensions, etc.) in its source code form (i.e., the FORTRAN statements). This source code must be converted into machine-usable instructions called object code. If many sets of data are to be run independently, so that no run is dependent on the successful completion of a previous run, the source code should be compiled once into the object code, which is called QUANTOC, and recompilation avoided for consecutive runs. In figure 1 the information flow to and from this object code is shown.

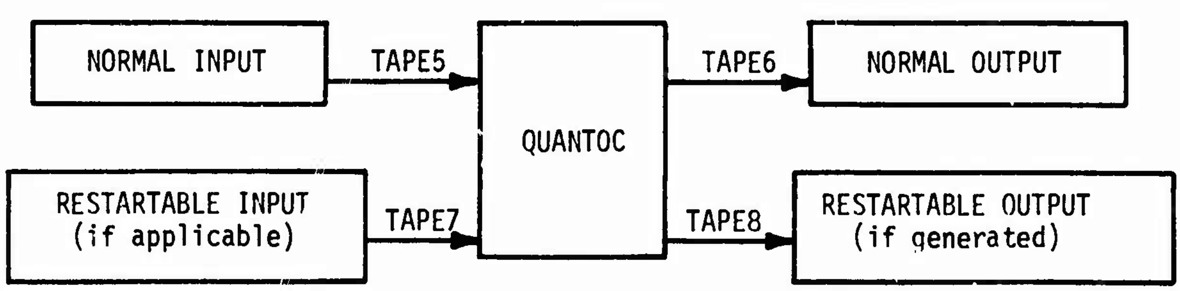

Figure 1. General Flow of Information

The identifiers TAPE5, TAPE6, TAPE7, and TAPE8 shown in figure 1 refer to input/output (I/O) devices which may be referenced by I/O unit numbers in the FORTRAN READ and WRITE statements in QUANTO. These four identifiers are listed in the first statement of QUANTO, the program header card, shown here:

```
PROGRAM QUANTO(INPUT,OUTPUT,TAPE7,TAPE8,TAPE5=INPUT,TAPE6=OUTPUT)
```

The "normal input' (figure 1) is data cards or "card images" from a data file. This card input is read from the TAPE5 input device. The amount of input required from this device depends on the MODE option to be used, as described in table 2.

"Normal output" (figure 1) is that produced by QUANTO during the processing of a job and is continually transmitted to an output buffer (disk storage), identified as TAPE6. This output is subsequently disposed to a printer or a microfilm output device.

"Restartable output" (figure 1) may be generated by certain jobs to permit a subsequent job to continue processing using the results of the first job as input. As described in table 2, all information necessary for the continuation of processing may be requested after the initial computations (survivabilities and expected kill) by inputing MODE=1 or may be obtained only on the condition that a job is within 30 seconds of its time limit (when MODE=2 or 3). The latter option prohibits jobs from terminating at the time limit without saving the results of the last computation cycle in a convenient form. The restartable output is transmitted to the output device indicated as TAPE8 and ultimately is saved on a physical magnetic tape. Jobs which create a restart tape may be continued by reading the created restart tape as input with a MODE=2 job. The restartable input (figure 1) is read from the magnetic tape input device identified as TAPE7.

The creation of restartable I/O has several purposes: (1) it prevents jobs from terminating at the time limit, due to an insufficient user time estimate, without saving the results of its computations; (2) it eliminates the necessity to reinitiate the problem with an increased time limit; and (3) it allows the user to analyze intermediate results to assure himself that the job is progressing properly before he risks a greater expenditure of computer time.

2. JOB CONTROL VARIATIONS

Job control language provides a means to exercise control over the manner in which a particular job is processed by a computer system. Information sources are directly (or indirectly by default) specified through control cards. The following examples offer various schemes for using the QUANTO program on the AFWL computer system. Bracketed upper-case Latin letters are used in line with the control cards to indicate information in punched cards, e.g., the deck containing the QUANTO source code. Each example is followed by a brief description of the overall process.

a. Example I

```
JOBID,P2, T1000,CM240000.
TASK(MILLER,00000000-0XX,ORGANIZATION,TELEPHONE)
REQUEST TAPE7.  XX000  RING OUT
REWIND(TAPE7)
REQUEST TAPE8.  XX001  RING IN
REWIND(TAPE8)
RUN(A,,,INPUT,OUTPUT,QUANTOC,377777)
PRESET.
QUANTOC(LC=377777,INPUT,OUTPUT,TAPE7,TAPE8)
REWIND(TAPE7)
RETURN(TAPE7)
REWIND(TAPE8)
RETURN(TAPE8)
EXIT.
DMP(240000)
7/8/9
[QUANTO SOURCE CODE PUNCHED CARD DECK]
7/8/9
[DATA PUNCHED CARD DECK]
6/7/8/9
```

Example I illustrates job control language which runs QUANTO in batch processing on the AFWL computer system. In the job card a core requirement of 240,000 (octal) words and a time limit of 1000 (octal) seconds are specified. These limits will be sufficient for most individual problems run by QUANTO. In particular, the core is sufficient for the array sizes indicated in table 18. As each tape request card is processed, a flashing message appears on the computer operator's console. The operator then mounts the tapes identified by XX000 and XX001. This example assumes the presence of restartable input (TAPE7) and the possibility of generating restartable output (TAPE8) which is to be saved. If either of these conditions is absent, all references to the corresponding tape should be removed from the job control language.

The RUN card requests compilation of the QUANTO source code into the object code file called QUANTOC. The LC=377777 in the line requesting execution of the QUANTOC code removes the line count limit on output. This is frequently required since QUANTO currently generates considerable output for many types of problems and values of the input parameters. A compiled listing of the program statements and the program would appear on paper using the job control language of example I. The DMP card requests a dump of the entire core segment if an abnormal termination of the job occurs. The notations "7/8/9" and "6/7/8/9" indicate cards having multiple numeric punches in column one which translate to "end of record" and "end of file," respectively, on magnetic tape.

A major problem with the previous example is that it ties up two tape-drive units for the duration of the program run. Example II affords a means of transferring information to and from QUANTO via disk storage, again assuming that both tapes (TAPE7, TAPE8) are necessary and contain information.

b. Example II

```
JOBID,P2,T1000,CM60000.
TASK(NAME,00000000-0XX,ORGANIZATION,TELEPHONE)
RFL(10000) REDUCE TO 10K FOR TAPE HANDLING
REQUEST TAPE7.  XX000  RING OUT
REWIND(TAPE7)
COPY(TAPE7,DISK7)
REWIND(TAPE7)
RETURN(TAPE7)
RFL(100000) FOR COMPILE
RUN(A,,,INPUT,OUTPUT,QUANTOC,377777)
RFL(240000) FOR LOAD
PRESET.
REWIND(DISK7)
QUANTOC(LC=377777,INPUT,FILMPR,DISK7,DISK8)
RFL(10000)
REQUEST TAPE8.  XX000  RING IN
REWIND(TAPE8)
REWIND(DISK8)
COPY(DISK8,TAPE8)
REWIND(TAPE8)
RETURN(TAPE8)
EXIT.
DMP(240000)
7/8/9
[QUANTO SOURCE CODE PUNCHED CARD DECK]
7/8/9
[DATA PUNCHED CARD DECK]
6/7/8/9
```

Note that the job card requests 60,000 (octal) words of core memory to conform with the AFWL priority system but that the needed core is adjusted throughout the job control language by RFL cards. The output from the execution of QUANTOC is placed on microfilm due to the FILMPR entry on the QUANTOC execute card.

A major drawback of the first two examples is the punched card handling, since the QUANTO source deck contains approximately 4500 cards. A method for handling large jobs is available under the AFWL computer system and is shown in the next example.

c. Example III

```
JOBID,P2,T1000,CM60000.
TASK(NAME,00000000-0XX,ORG,TEL)
RFL(10000) REDUCE TO 10K FOR TAPE HANDLING
REQUEST TAPE7.  XX000  RING OUT
REWIND(TAPE7)
COPY(TAPE7,DISK7)
REWIND(TAPE7)
RETURN(TAPE7)
COMMON(SABCEM)
RFL(40000) FOR UPDATE
UPDATE(Q,P=SABCEM,D,L=A1)
RETURN(SABCEM)
RFL(100000) FOR COMPILE
RUN(A,,,COMPILE,OUTPUT,QUANTOC,377777)
RFL(240000) FOR LOAD
PRESET.
REWIND(DISK7)
QUANTOC(LC=377777,INPUT,FILMPR,DISK7,DISK8)
RFL(10000)
REQUEST TAPE8.  XX001  RING IN
REWIND(TAPE8)
REWIND(DISK8)
COPY(DISK8,TAPE8)
REWIND(TAPE8)
RETURN(TAPE8)
7/8/9
*COMPILE QUANTOT
7/8/9
[DATA PUNCHED CARD DECK]
6/7/8/9
```

For this example, the QUANTO source code is assumed to be previously built as an update file called QUANTOT and placed in the COMMON file, SABCEM. The AFWL update system could create temporary changes to the QUANTOT file if these changes appeared after the *COMPILE QUANTOT card; for instance, array sizes could be changed by replacing the appropriate DIMENSION statements. Such

changes, if any, are made when the UPDATE card is encountered in the job control language. The resultant updated program, in source code form, is placed in a file named COMPILE by the UPDATE step. Thus, COMPILE appears in the RUN card instead of the usual INPUT.

For production purposes the object code QUANTOC could be saved on disk as shown in example IV. Then multiple executions of QUANTOC could be requested for multiple sets of data without requiring multiple compilations of the source code.

d. Example IV

```
JOBID,P5,T177,CM40000.
TASK(NAME,00000000-OXX,ORGANIZATION,TELEPHONE)
COMMON(SABCEM)
UPDATE(Q,P=SABCEM,L=A1,D)
RETURN(SABCEM)
RUN(A,,,COMPILE,OUTPUT,QUANTOC,377777)
COMMON(QUANTOC)
RETURN(QUANTOC)
7/8/9
*COMPILE QUANTOT
7/8/9
6/7/8/9
```

Obviously many variations of job control language are possible, but the preceding examples should suffice to demonstrate the basic means of running QUANTO.

SECTION IV

SUMMARY

This report has described the input format, array dimensions, and job control language of concern to the user of the program QUANTO. It should be expected that occasional problems with the program will occur as the user attemps runs with new combinations of parameter values. Such difficulties should be brought to the attention of the authors, who will advise and instruct the user of QUANTO, or the Air Force Weapons Laboratory. As QUANTO is used, modifications and improvements are inevitable. The most current version of QUANTO and the documentation may be obtained from the Air Force Weapons Laboratory.

The mathematical model used by QUANTO would be appropriate for applications other than SLBM attacks on flushing aircraft, with slight modifications. For instance, if the survivability of any target from any weapon can be quantified, a weapon allocation may be obtained from the optimization model. A complete description of the mathematical model and its assumptions may be found in AFWL Technical Report AFWL-TR-73-242.

APPENDIX I

CURRENT LISTING OF QUANTO SOURCE CODE

```
      PROGRAM QUANTO(INPUT,OUTPUT,TAPE7,TAPE8,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION ALOC(35,168), FLAMB(35,168),
     1 SURV(35,168,2), FLAMBI(35,168,2),
     2 TGTLAT(35),TGTLNG(35),DTCENT(35),NRWAYS(35),NWALOC(35),
     3 VAL(35,2),PROD(35,2),VKILL(35,2),
     4 ISEQ(35,34),BRT(34),
     5 RELVAL(2),BRTIME(2),PSI(2),ICAL(2),PKCIR(2),PKAN(2),
     6 SUBLAT(28),SUBLNG(28),ISUBS(28),NMPS(28),MTYPE(28),ITGTNO(16),
     7 DELTH(4),RELML(4),RELMF(4),RELMWH(4),RNGMAX(4),RNGMIN(4),
     8 YIELD(4),FMTIME(15,4),FMRNG(15,4),MPROF(4),
     9 QPTS(30),QAREAL(30,2,4),QRLMX(30,2,4),DELTAC(2,2,2),JFLAC(2,2)
      COMMON /DISTIME/ FOTIME(99,2),FORNG(99,2),NPROF(2),CV(2),TI(2),JTY
     1PE,NTYPES,MISTYP,MTYPES,RLMAX
      DATA RDPDEG,VERYHI,PI/.0174532925,1.0E+300,3.14159265/
      DATA MXTGT,MXWPNS/35,168/
      EQUIVALENCE (FLAMB,FLAMBI)
C     IOUT=1 SUPPRESSES INTERMEDIATE MULTIPLIER AND ALLOCATION OUTPUT.
C     OTHERWISE SET IOUT=2.
C     IF ISOPT=1, SUB LOCATIONS WILL NOT BE OPTIMIZED AMONG INPUT SUB
C     LOCATIONS.
C     IF ISOPT=2, SUB LOCATIONS WILL BE OPTIMIZED.
C     IF IVOPT = 1, BEDDOWN WILL NOT BE OPTIMIZED.
C     IF IVOPT = 2, BEDDOWN WILL BE OPTIMIZED.
10    READ 890, NTGTS,NSUBS,NTYPES,MXRWAY,MTYPES,IOUT,ISOPT,IVOPT,NCASE,
     1MODE
      IF (ENDFILE 5) 800,20
20    CALL DATE (IDATE)
      CALL CLOCK (ITIME)
      CALL PAGE (0)
      PRINT 900, NCASE,MODE,IDATE,ITIME
      PRINT 910, NTGTS,NSUBS,NTYPES,MTYPES
      CALL PAGE (2)
      IF (MODE.NE.2) GO TO 30
      READ (7) NWPNS,NCOL,ITER,ITCUT1,ITCUT2,MOVES,MOVEV,MOVEST,EPS,EPSC
     1UT,OBHOLD,DELOBJ,SUBOBJ,CHGKIL
      READ (7) (((SURV(I,J,K),I=1,NTGTS),J=1,NWPNS),K=1,NTYPES)
      READ (7) ((ALOC(I,J),I=1,NTGTS),J=1,NWPNS)
      READ (7) ((VAL(I,J),I=1,NTGTS),J=1,NTYPES),(RELVAL(I),I=1,NTYPES),
     1(ISUBS(I),NMPS(I),MTYPE(I),I=1,NSUBS)
      PRINT 920
      CALL PAGE (1)
      GO TO 610
30    DO 50 I=1,NTGTS
      READ 930, (TGTLAT(I),TGTLNG(I),DTCENT(I),NRWAYS(I),(VAL(I,JTYPE),J
     1TYPE=1,NTYPES))
      NAC=0
      DO 40 JTYPE=1,NTYPES
40    NAC=NAC+VAL(I,JTYPE)+.01
50    READ 940, (ISEQ(I,IAC),IAC=1,NAC)
      READ 950, (RELVAL(JTYPE),BRTIME(JTYPE),PSI(JTYPE),ICAL(JTYPE),JTYP
     1E=1,NTYPES)
```

```
      READ 960, (((DELTAC(IRW,JTYPE,KTYPE),KTYPE=1,NTYPES),JTYPE=1,NTYPE
     1S),IRW=1,MXRWAY)
      PRINT 970
      CALL PAGE (3)
      DO 70 I=1,NTGTS
      NAC=0
      DO 60 JTYPE=1,NTYPES
60    NAC=NAC+VAL(I,JTYPE)+.01
      FNAC=NAC
      NLINES=(FNAC-1.)/30.
      CALL PAGE (-MAX0(2,NTYPES)-NLINES-2)
      PRINT 980, I,TGTLAT(I),TGTLNG(I),NRWAYS(I),DTCENT(I),(JTYPE,VAL(I,
     1JTYPE),JTYPE=1,NTYPES)
70    PRINT 990, (ISEQ(I,IAC),IAC=1,NAC)
      CALL PAGE (-NTYPES-3)
      PRINT 1000, (JTYPE,RELVAL(JTYPE),BRTIME(JTYPE),PSI(JTYPE),ICAL(JTY
     1PE),JTYPE=1,NTYPES)
      CALL PAGE (-MXRWAY*NTYPES*NTYPES-3)
      PRINT 1010, (((IRW,JTYPE,KTYPE,DELTAC(IRW,JTYPE,KTYPE),KTYPE=1,NTY
     1PES),JTYPE=1,NTYPES),IRW=1,MXRWAY)
      NWPNS=0
      NCOL=0
      DO 80 I=1,NSUBS
      READ 1020, SUBLAT(I),SUBLNG(I),ISUBS(I),NMPS(I),MTYPE(I)
      NWPNS=NWPNS+NMPS(I)
80    IF (ISUBS(I).NE.0) NCOL=NCOL+NMPS(I)
      CALL PAGE (-NSUBS-2)
      PRINT 1030, (I,SUBLAT(I),SUBLNG(I),ISUBS(I),NMPS(I),MTYPE(I),I=1,N
     1SUBS)
      CALL PAGE (-MTYPES*2)
      DO 100 J=1,MTYPES
      NSUBT=0
      DO 90 I=1,NSUBS
90    IF (MTYPE(I).EQ.J) NSUBT=NSUBT+ISUBS(I)
100   PRINT 1040, J,NSUBT
      DO 110 I=1,MTYPES
      READ 960, DELTM(I),RELML(I),RELMF(I),RELMWH(I),RNGMIN(I),RNGMAX(I)
     1,YIELD(I)
      READ 1050, MPR,(FMTIME(J,I),FMRNG(J,I),J=1,MPR)
110   MPROF(I)=MPR
      CALL PAGE (-MTYPES-3)
      PRINT 1060, (JTYPE,DELTM(JTYPE),RELML(JTYPE),RELMF(JTYPE),RELMWH(J
     1TYPE),RNGMIN(JTYPE),RNGMAX(JTYPE),YIELD(JTYPE),JTYPE=1,MTYPES)
      DO 120 JTYPE=1,MTYPES
      MPR=MPROF(JTYPE)
      CALL PAGE (-MPR-2)
120   PRINT 1070, JTYPE,(FMTIME(I,JTYPE),FMRNG(I,JTYPE),I=1,MPR)
      CALL PROCESS (PSI,ICAL,YIELD,NCASE,MODE)
      IF (MODE.EQ.0) GO TO 10
C
C
```

```
C     COMPUTE SURVIVABILITIES.
C     CONVERT LOCATION DEGREES TO RADIANS.
      DO 130 I=1,NTGTS
      TGTLAT(I)=TGTLAT(I)*RDPDEG
130   TGTLNG(I)=TGTLNG(I)*RDPDEG
      DO 140 I=1,NSUBS
      SUBLAT(I)=SUBLAT(I)*RDPDEG
140   SUBLNG(I)=SUBLNG(I)*RDPDEG
C     SET-UP FOR BUILDING TABLES OF LETHAL AREAS.
      DO 150 IQ=1,30
      IF (IQ.LE.11) QPTS(IQ)=IQ-1.
150   IF (IQ.GT.11) QPTS(IQ)=(IQ-9.)*5.
      DTCOLD=-1.
      DO 530 I=1,NTGTS
      CPKTIM=VERYHI
      DTCNEW=DTCENT(I)
      IF (DTCNEW.EQ.DTCOLD) GO TO 180
      DTCOLD=DTCNEW
C     GENERATE TABLE OF LETHAL AREAS FOR NEW DISTANCE TO CENTROID.
      DO 160 IQ=1,30
      DO 160 JTYPE=1,NTYPES
      DO 160 MISTYP=1,MTYPES
      QAREAL(IQ,JTYPE,MISTYP)=DETAREA(QPTS(IQ),DTCNEW)
160   QRLMX(IQ,JTYPE,MISTYP)=RLMAX
      CALL PAGE (0)
      PRINT 1080, DTCNEW
      CALL PAGE (1)
      DO 170 JTYPE=1,NTYPES
      DO 170 MISTYP=1,MTYPES
      CALL PAGE(-33)
170   PRINT 1090, JTYPE,MISTYP,(QPTS(IQ),QAREAL(IQ,JTYPE,MISTYP),QRLMX(I
     1Q,JTYPE,MISTYP),IQ=1,30)
      CALL PAGE (0)
180   NAC=0
      DO 190 JTYPE=1,NTYPES
190   NAC=NAC+VAL(I,JTYPE)+.01
C     COMPUTE BRAKE RELEASE TIMES OF EACH AIRCRAFT.
      ITYPE=ISEQ(I,1)
      BRT(1)=BRTIME(ITYPE)
      IF (NAC.LT.2) GO TO 210
      DO 200 IAC=2,NAC
      NWY=NRWAYS(I)
      ITYP1=ISEQ(I,IAC-1)
      ITYPE=ISEQ(I,IAC)
200   BRT(IAC)=BRT(IAC-1)+DELTAC(NWY,ITYP1,ITYPE)
      CALL PAGE (-NAC-2)
210   PRINT 1100, I,(ISEQ(I,IAC),BRT(IAC),IAC=1,NAC)
C     LOCATE FIRST AND LAST AIRCRAFT OF EACH TYPE.
      DO 220 IAC=1,2
      DO 220 JTYPE=1,NTYPES
220   JFLAG(IAC,JTYPE)=0
```

```
      DO 270 JTYPE=1,NTYPES
      DO 230 IAC=1,NAC
      JAC=IAC
      IF (ISEQ(I,JAC).EQ.JTYPE) GO TO 240
230   CONTINUE
C     NO AIRCRAFT OF THIS TYPE FOUND.
      GO TO 270
240   JFLAC(1,JTYPE)=JAC
      DO 250 IAC=1,NAC
      JAC=NAC+1-IAC
      IF (ISEQ(I,JAC).EQ.JTYPE) GO TO 260
250   CONTINUE
C     WE WILL BE ABLE TO FIND ONE OF THIS TYPE IF WE GET TO THIS POINT,
C     BUT ANYWAY...
      GO TO 270
260   JFLAC(2,JTYPE)=JAC
270   CONTINUE
C     COMPUTE TIMES OF FLIGHT AND ARRIVALS IN MINUTES.  IF DISTANCE IS
C     OUT OF RANGE, MAKE FLIGHT TIME VERY LARGE.
      K=0
      DO 520 KSUB=1,NSUBS
      DISTT=DIST(TGTLAT(I),TGTLNG(I),SUBLAT(KSUB),SUBLNG(KSUB))
      MISTYP=MTYPE(KSUB)
      RELM=RELML(MISTYP)*RELMF(MISTYP)*RELMWH(MISTYP)
      IF (DISTT.LT.RNGMIN(MISTYP).OR.DISTT.GT.RNGMAX(MISTYP)) GO TO 280
      TFLT=ALAG(DISTT,FMRNG(1,MISTYP),FMTIME(1,MISTYP),MPROF(MISTYP))
      GO TO 290
280   TFLT=VERYHI
290   PRINT 1110, I,KSUB,DISTT,MISTYP,TFLT
      CALL PAGE (2)
      NMIS=NMPS(KSUB)
      DO 520 KSAL=1,NMIS
      K=K+1
      TOAT=TFLT+(KSAL-1)*DELTM(MISTYP)
      IF (TOAT.LT.VERYHI) GO TO 310
      DO 300 JTYPE=1,NTYPES
300   SURV(I,K,JTYPE)=1.
      GO TO 520
310   RFC1=0.
      RFCL=9999.
C     COMPUTE RANGE OF FIRST AIRCRAFT AND LAST IF  PK  SWITCH INDICATES
C     ANNULAR PK COMPUTATION MAY STILL BE NECESSARY. (TOAT.LE.CPKTIM)
      DO 320 JTYPE=1,NTYPES
      JAC=JFLAC(1,JTYPE)
      IF (JAC.EQ.0) GO TO 320
      RTOA=TOAT-BRT(JAC)
      RFC=PROFLU(RTOA,FOTIME(1,JTYPE),FORNG(1,JTYPE),NPROF(JTYPE))-DTCEN
     1T(I)
      IF (RFC.GT.RFC1) RFC1=RFC
C     COMPUTE ONLY RADIUS OF FIRST AIRCRAFT IF SWITCH FOR CIRCULAR PK
C     COMPUTATION IS SET.
```

```
      IF (TOAT.GE.CPKTIM) GO TO 320
      JAC=JFLAC(2,JTYPE)
      RTOA=TOAT-BRT(JAC)
      RFC=PROFLU(RTOA,FOTIME(1,JTYPE),FORNG(1,JTYPE),NPROF(JTYPE))-DTCEN
     1T(I)
      IF (RFC.LT.RFCL) RFCL=RFC
  320 CONTINUE
      IF (RFCL.GE.RFC1) RFCL=RFC1-.01
      IF (RFCL.LE.0.) RFCL=0.
      IF (KSAL.NE.1) GO TO 330
      PRINT 1120, K,I,RFC1
      CALL PAGE (1)
  330 RFC1SQ=RFC1*RFC1
      RFCLSQ=RFCL*RFCL
      ISUM=0
      IF (TOAT.GE.CPKTIM) GO TO 400
C     COMPUTE LARGEST LETHAL RADIUS FOR CENTERED WEAPON.
      CELRMX=0.
      DO 340 JTYPE=1,NTYPES
      CELR=SQRT(QAREAL(1,JTYPE,MISTYP)/PI)
  340 IF (CELR.GT.CELRMX) CELRMX=CELR
      IF (RFC1.GT.CELRMX) GO TO 380
      IF (RFC1.GT.0.) GO TO 360
      DO 350 JTYPE=1,NTYPES
  350 SURV(I,K,JTYPE)=1.-RELM
      GO TO 520
  360 DO 370 JTYPE=1,NTYPES
      PK=(QAREAL(1,JTYPE,MISTYP)-PI*RFCLSQ)/(PI*(RFC1SQ-RFCLSQ))
      IF (PK.GT.1.) PK=1.
      IF(PK.LT.0.) PK=0.
  370 SURV(I,K,JTYPE)=1.-PK*RELM
      GO TO 520
C     COMPUTE FARTHEST REACH OF LETHAL REGION FOR WEAPON PLACED AT
C     RFC1/SQRT(2) TO SEE IF IT PROTRUDES BEYOND RFC1.
  380 RLMX=0.
      DO 390 JTYPE=1,NTYPES
      RL=ALAG(RFC1/1.414213562,QPTS,QRLMX(1,JTYPE,MISTYP),30)
  390 IF (RL.GT.RLMX) RLMX=RL
      IF (RFC1/1.414213562+RLMX.GT.RFC1) GO TO 560
      ISUM=ISUM+1
C     COMPUTE CIRCULAR PKS.
  400 DO 410 JTYPE=1,NTYPES
      AREAL=ALAG(RFC1/1.414213562,QPTS,QAREAL(1,JTYPE,MISTYP),30)
      IF (AREAL.LT.0.) AREAL=0.
  410 PKCIR(JTYPE)=AREAL/(PI*RFC1SQ)
      IF (TOAT.GE.CPKTIM) GO TO 480
C     COMPUTE ANNULAR PKS UNTIL A LARGER ONE IS FOUND.
  420 DO 450 JTYPE=1,NTYPES
      AREAL=ALAG((RFC1+RFCL)/2.,QPTS,QAREAL(1,JTYPE,MISTYP),30)
      IF (AREAL.LT.0.) AREAL=0.
      ELR=SQRT(AREAL/PI)
```

```
      IF (ELR.LT..0001) GO TO 430
      PKAN(JTYPE)=XAREA(ELR,(RFC1+RFCL)/2.,RFCL,RFC1)/(PI*(RFC1SQ-RFCLSQ
     1))
      GO TO 440
430   PKAN(JTYPE)=0.
440   PRINT 1130, PKAN(JTYPE),PKCIR(JTYPE),K,JTYPE,RFC1,RFCL
      IF (PKAN(JTYPE).LT.0.) STOP
      CALL PAGE (1)
      IF (PKCIR(JTYPE).LE.PKAN(JTYPE)) GO TO 470
450   CONTINUE
C     USE ANNULAR PKS.
      DO 450 JTYPE=1,NTYPES
460   SURV(I,K,JTYPE)=1.-PKAN(JTYPE)*RELM
      GO TO 520
470   ISUM=ISUM+1
480   DO 490 JTYPE=1,NTYPES
      PK=AMIN1(1.0,PKCIR(JTYPE))
490   SURV(I,K,JTYPE)=1.-PK*RELM
      IF (ISUM.EQ.2.AND.TOAT.LT.CPKTIM) CPKTIM=TOAT
      GO TO 520
500   DO 510 JTYPE=1,NTYPES
      QPOS=AMAX1(0.,RFC1-RLMX)
      AREAL=ALAG(QPOS,QPTS,QAREAL(1,JTYPE,MISTYP),30)
      IF (AREAL.LT.0.) AREAL=0.
510   PKCIR(JTYPE)=AREAL/(PI*RFC1SQ)
      GO TO 420
520   CONTINUE
530   CONTINUE
      DO 540 JTYPE=1,NTYPES
      CALL PAGE (0)
      CALL PAGE (2)
      PRINT 1140, JTYPE
      DO 540 I=1,NTGTS
      CALL PAGE (-NSUBS-2)
      PRINT 1150, I
      JLO=1
      DO 540 JJ=1,NSUBS
      JHI=JLO+NMPS(JJ)-1
      PRINT 1160, (SURV(I,J,JTYPE),J=JLO,JHI)
540   JLO=JHI+1
      SUBOBJ=0.
      MOVES=0
      MOVEST=0
      MOVEV=0
      READ 1170, CHGKIL,ITCUT1,ITCUT2,EPSCUT
      EPS=.1
      PRINT 1180, EPS
      ITER=0
      CALL PAGE (3)
C     COMPUTE VALUES OF AIRCRAFT.
      DO 550 I=1,NTGTS
```

```
      DO 550 JTYPE=1,NTYPES                                             QUA3060
550   VAL(I,JTYPE)=VAL(I,JTYPE)*RELVAL(JTYPE)                           QUA3070
      DO 560 I=1,NTGTS                                                  QUA3080
      DO 560 J=1,NWPNS                                                  QUA3090
560   ALOC(I,J)=0.                                                      QUA3100
C     INPUT FIRST ALLOCATION.                                           QUA3110
      MWPN=0                                                            QUA3120
      DO 600 J=1,NSUBS                                                  QUA3130
      JLIM=ISUBS(J)                                                     QUA3140
      ILIM=NMPS(J)                                                      QUA3150
      IF (JLIM.EQ.0) GO TO 590                                          QUA3160
      DO 580 JSUB=1,JLIM                                                QUA3170
      READ 890, (ITGTNO(I),I=1,ILIM)                                    QUA3180
      DO 570 I=1,ILIM                                                   QUA3190
      MWPN=MWPN+1                                                       QUA3200
      II=ITGTNO(I)                                                      QUA3210
      IF (II.EQ.0) GO TO 570                                            QUA3220
      ALOC(II,MWPN)=ALOC(II,MWPN)+1.0                                   QUA3230
570   CONTINUE                                                          QUA3240
      IF (JSUB.LT.JLIM) MWPN=MWPN-ILIM                                  QUA3250
580   CONTINUE                                                          QUA3260
      GO TO 600                                                         QUA3270
590   MWPN=MWPN+ILIM                                                    QUA3280
600   CONTINUE                                                          QUA3290
C                                                                       QUA3300
C                                                                       QUA3310
C     LOOP ON ALLOCATIONS FOLLOWS.                                      QUA3320
610   IBROUT=0                                                          QUA3330
      CALL TGTKIL (NTGTS,NWPNS,NTYPES,OBJSUM,ALOC,FLAMB,FLAMBI,SURV,VAL,QUA3340
     1PROD,VKILL,NWALOC,RELVAL,MXTGT,MXWPNS)                            QUA3350
      PRINT 1190, OBJSUM                                                QUA3360
      CALL PAGE (2)                                                     QUA3370
      IF (MODE.EQ.1) GO TO 620                                          QUA3380
      CALL TIMTGO (TLEFT)                                               QUA3390
      IF (TLEFT.GT.30.) GO TO 630                                       QUA3400
620   WRITE (8) NWPNS,NCOL,ITER,ITCUT1,ITCUT2,MOVES,MOVEV,MOVEST,EPS,EPSQUA3410
     1CUT,OBHOLD,DELOBJ,SUBOBJ,CHGKIL                                   QUA3420
      WRITE (8) (((SURV(I,J,K),I=1,NTGTS),J=1,NWPNS),K=1,NTYPES)        QUA3430
      WRITE (8) ((ALOC(I,J),I=1,NTGTS),J=1,NWPNS)                       QUA3440
      WRITE (8) ((VAL(I,J),I=1,NTGTS),J=1,NTYPES),(RELVAL(I),I=1,NTYPES)QUA3450
     1,(ISUBS(I),NMPS(I),MTYPE(I),I=1,NSUBS)                            QUA3460
      PRINT 1200                                                        QUA3470
      CALL PAGE (1)                                                     QUA3480
      IF (MODE.EQ.1) GO TO 10                                           QUA3490
      STOP                                                              QUA3500
630   OBHOLD=0.                                                         QUA3510
C     MAX NUMBER OF ITERATIONS IS THE PRODUCT OF UPPER LIMITS OF        QUA3520
C     ITER1 AND ITER2.                                                  QUA3530
      IF (ITCUT1.EQ.0) GO TO 730                                        QUA3540
      DO 720 ITER1=1,ITCUT1                                             QUA3550
      DO 700 ITER2=1,ITCUT2                                             QUA3560
```

```
      ITER=ITER+1
      CALL ADJLAM (NTGTS,NSUBS,NTYPES,NCOL,IBROUT,MLOW,MHI,JCOL,DELTA,EP
     1S,ALOC,FLAMB,FLAMBI,SURV,NWALOC,ISUBS,NMPS,MXTGT,MXWPNS)
      IF (IBROUT.NE.1) GO TO 660
      IF (EPS.GT.1.5*EPSCUT) GO TO 640
      PRINT 1210, EPSCUT
      CALL PAGE (2)
      GO TO 730
  640 EPS=EPS/10.
      PRINT 1180, EPS
      CALL PAGE (3)
      IBROUT=0
      IF (IOUT.EQ.1) GO TO 700
C     DEBUG OUTPUT.
      CALL ALOUT (ITER,NTGTS,NSUBS,NWPNS,ALOC,ISUBS,NMPS,MXTGT)
      CALL PAGE (0)
      CALL PAGE (3)
      PRINT 1220
      DO 650 I=1,NTGTS
      CALL PAGE (-NSUBS-2)
      PRINT 1150, I
      JLO=1
      DO 650 JJ=1,NSUBS
      JHI=JLO+NMPS(JJ)-1
      PRINT 1160, (FLAMB(I,J),J=JLO,JHI)
  650 JLO=JHI+1
      GO TO 700
C     CHANGE THE PRODUCT MATRIX, VALUE KILLED ON EACH TARGET, AND
C     MULTIPLIER MATRIX AS CHANGED BY CHANGED ALLOCATION.
  660 DO 670 JTYPE=1,NTYPES
      SNEG=SURV(MLOW,JCOL,JTYPE)**(-DELTA)
      SPOS=SURV(MHI,JCOL,JTYPE)**DELTA
      PROD(MLOW,JTYPE)=PROD(MLOW,JTYPE)*SNEG
      PROD(MHI,JTYPE)=PROD(MHI,JTYPE)*SPOS
      OBJSUM=OBJSUM-VKILL(MLOW,JTYPE)-VKILL(MHI,JTYPE)
      VKILL(MLOW,JTYPE)=VAL(MLOW,JTYPE)*(1-PROD(MLOW,JTYPE))
      VKILL(MHI,JTYPE)=VAL(MHI,JTYPE)*(1-PROD(MHI,JTYPE))
      OBJSUM=OBJSUM+VKILL(MLOW,JTYPE)+VKILL(MHI,JTYPE)
      DO 670 J=1,NWPNS
      FLAMBI(MLOW,J,JTYPE)=FLAMBI(MLOW,J,JTYPE)*SNEG
  670 FLAMBI(MHI,J,JTYPE)=FLAMBI(MHI,J,JTYPE)*SPOS
      DO 690 J=1,NWPNS
      SUM1=0.
      SUM2=0.
      DO 680 JTYPE=1,NTYPES
      SUM1=SUM1+FLAMBI(MLOW,J,JTYPE)
  680 SUM2=SUM2+FLAMBI(MHI,J,JTYPE)
      FLAMB(MLOW,J)=SUM1
  690 FLAMB(MHI,J)=SUM2
      IF (IOUT.EQ.1) GO TO 700
      CALL PAGE (-7)
```

```
      PRINT 1230, ITER
      PRINT 1240, JCOL,MLOW,MHI,DELTA
      PRINT 1190, OBJSUM
700   CONTINUE
      IF (IOUT.NE.1) GO TO 710
      CALL PAGE (-5)
      PRINT 1230, ITER
      PRINT 1190, OBJSUM
710   DELOBJ=OBJSUM-OBHOLD
      OBHOLD=OBJSUM
      IF (DELOBJ.GE.CHGKIL) GO TO 720
      PRINT 1250, DELOBJ,CHGKIL,ITCUT2
      CALL PAGE (2)
      GO TO 730
720   CONTINUE
C     BRANCH OUT IF NO MORE ITERATIONS ARE POSSIBLE.
      PRINT 1260
      CALL PAGE (2)
      GO TO 730
730   PRINT 1270, EPS
      CALL PAGE (2)
      CALL ALOUT (ITER,NTGTS,NSUBS,NWPNS,ALOC,ISUBS,NMPS,MXTGT)
      CALL KILOUT (NTGTS,NWPNS,NTYPES,OBJSUM,ALOC,FLAMB,FLAMBI,SURV,VAL,
     1PROD,VKILL,NWALOC,RELVAL,MXTGT,MXWPNS)
      PRINT 1190, OBJSUM
      CALL PAGE (2)
      IF (ISOPT.NE.1) GO TO 610
C
C
C     POST-PROCESSING TO RELOCATE AIRCRAFT.
740   IF (IVOPT.EQ.1) GO TO 780
      PSUM=0.
      DO 770 JTYPE=1,NTYPES
      PMIN=2.0
      PMAX=-1.0
      CALL PAGE (0)
      CALL PAGE (-NTGTS)
      DO 760 I=1,NTGTS
      PTEST=PROD(I,JTYPE)
      PRINT 1280, I,JTYPE,PTEST
      IF (PTEST.GE.PMIN.OR.VAL(I,JTYPE).LE..0001) GO TO 750
      PMIN=PTEST
      IPMIN=I
750   IF (PTEST.LE.PMAX) GO TO 760
      PMAX=PTEST
      IPMAX=I
760   CONTINUE
      VMIN=VAL(IPMIN,JTYPE)
      PDELT=VMIN*(PMAX-PMIN)
      IF (VMIN.LT..1.OR.PDELT.GT.VMIN) PDELT=VMIN
      PSUM=PSUM+PDELT
```

```
      VAL(IPMAX,JTYPE)=VAL(IPMAX,JTYPE)+PDELT
      VAL(IPMIN,JTYPE)=VMIN-PDELT
      PDELT=PDELT/RELVAL(JTYPE)
770   PRINT 1290, PDELT,JTYPE,IPMIN,IPMAX
      MOVEV=MOVEV+1
      PRINT 1300, MOVEV
      CALL PAGE (2)
      IF (PSUM.LT..05) GO TO 780
      EPS=.1
      PRINT 1180, EPS
      CALL PAGE (3)
      GO TO 610
C
C
C     WRAP-UP INTEGERIZATION AND OUTPUT.
780   CALL ALINT (NTGTS,NSUBS,ALOC,ISUBS,NMPS,MXTGT)
      PRINT 1310
      CALL PAGE (1)
      IF (IVOPT.NE.2) GO TO 790
      CALL VINT (NTGTS,NTYPES,VAL,RELVAL,MXTGT)
      PRINT 1320
      CALL PAGE (1)
790   CALL ALOUT (ITER,NTGTS,NSUBS,NWPNS,ALOC,ISUBS,NMPS,MXTGT)
      CALL TGTKIL (NTGTS,NWPNS,NTYPES,OBJSUM,ALOC,FLAMB,FLAMBI,SURV,VAL,
     1PROD,VKILL,NWALOC,RELVAL,MXTGT,MXWPNS)
      PRINT 1190, OBJSUM
      CALL PAGE (2)
      GO TO 10
800   PRINT 1330
      STOP
C
C
C     POST-PROCESSING TO RE-LOCATE SUBS.
810   IF (OBJSUM.GE.SUBOBJ) GO TO 830
      MOVES=MOVES+1
      IF (MOVES.LT.1) GO TO 840
820   ISOPT=1
      GO TO 740
830   MOVES=0
840   SUBOBJ=OBJSUM
      IF (IOUT.EQ.1) GO TO 860
      CALL PAGE (0)
      CALL PAGE (3)
      PRINT 1220
      DO 850 I=1,NTGTS
      CALL PAGE (-NSUBS-2)
      PRINT 1150, I
      JLO=1
      DO 850 JJ=1,NSUBS
      JHI=JLO+NMPS(JJ)-1
      PRINT 1160, (FLAMB(I,J),J=JLO,JHI)
```

```
850   JLO=JHI+1
860   IBOUT=0
      CALL SUBADJ (NTGTS,NSUBS,MTYPES,EPS,IBOUT,ALOC,FLAMB,ISUBS,NMPS,MT
     1TYPE,MXTGT)
      IF (IBOUT.EQ.1) GO TO 820
      MOVEST=MOVEST+1
      PRINT 1340, MOVEST
      CALL PAGE (1)
      IF (IOUT.EQ.1) GO TO 870
      CALL ALOUT (0,NTGTS,NSUBS,NWPNS,ALOC,ISUBS,NMPS,MXTGT)
870   NCOL=0
      DO 880 I=1,NSUBS
880   IF (ISUBS(I).GT.0) NCOL=NCOL+NMPS(I)
      CALL PAGE (-NSUBS-2)
      PRINT 1350, (I,ISUBS(I),MTYPE(I),I=1,NSUBS)
      EPS=.1
      PRINT 1130, EPS
      CALL PAGE (3)
      GO TO 610
C
890   FORMAT (14I5)
900   FORMAT (6H CASE ,I5,10X,5HMODE ,I5,10X,5HDATE ,A10,10X,5HTIME ,A10
     1)
910   FORMAT (17H NEW PROBLEM WITH,I5,9H TARGETS,,I5,15H SUB LOCATIONS,,
     1I5,23H TYPES OF AIRCRAFT, AND,I5,19H TYPES OF MISSILES.)
920   FORMAT (26H RESTART INFORMATION READ.)
930   FORMAT (3F10.4,I5,5X,3F10.4/(7F10.4))
940   FORMAT (70I1)
950   FORMAT (3F10.4,I5)
960   FORMAT (7F10.4)
970   FORMAT (/27H TARGET LOCATIONS (DEGREES),3X,7HRUNWAYS,2X,8HCENTROID
     1,2X,8HAIRCRAFT/39X,6HDISTANCE,6X,15HTYPE AND NUMBER)
980   FORMAT (/I7,2F10.4,I10,F10.4,I10,F11.4/(47X,I10,F11.4))
990   FORMAT (32X,25HTAKE-OFF SEQUENCE BY TYPE,5X,30I2/(62X,30I2))
1000  FORMAT (/2X,8HAIRCRAFT,2X,8HRELATIVE,5X,5HBRAKE,/6X,4HTYPE,5X,5HVA
     1LUE,3X,7HRELEASE,7X,3HPSI,7X,3HCAL,/(I10,3F10.4,I10))
1010  FORMAT (/28H AIRCRAFT TAKE-OFF INTERVALS/8X,7HRUNWAYS,5X,5HTYPE1,5
     1X,5HTYPE2,3X,7HMINUTES/(I15,2I10,F10.4))
1020  FORMAT (2F10.4,3I5)
1030  FORMAT (/2X,23HSUB LOCATIONS (DEGREES),6X,4HSUBS,3X,13HMISSILES AN
     1D ,4HTYPE/(I5,2F10.4,I10,I11,I8))
1040  FORMAT (/23H NUMBER OF SUBS OF TYPE,I5,2H =,I5)
1050  FORMAT (I5,5X,6F10.4/(6F10.4))
1060  FORMAT (/3X,7HMISSILE,4X,6HLAUNCH,8X,13HRELIABILITIES,16X,3HMIN,7X
     1,3HMAX,5X/6X,4HTYPE,2X,8HINTERVAL,5X,6HLAUNCH,2X,6HFLIGHT,1X,7HWAR
     2HEAD,3X,2(5X,5HRANGE),5X,5HYIELD/(I10,F10.4,3X,3F8.4,3X,3F10.3))
1070  FORMAT (/13H MISSILE TYPE,11X,4HTIME,5X,5HRANGE/I13,5X,2F10.4/(18X
     1,2F10.4))
1080  FORMAT (16H FOR DISTANCE OF,F6.2,30H NM TO CENTROID FROM START OF
     1,13HTAKE-OFF ROLL)
1090  FORMAT (/14H AIRCRAFT TYPE,I5,20H VERSUS MISSILE TYPE,I5//(19H WHE
```

```
     1N DETONATION IS,F7.2,32H NM FROM CENTROID, LETHAL AREA =,F8.2,22H
     2SQUARE NM AND EXTENDS,F6.2,26H NM FARTHER FROM CENTROID.))
1100  FORMAT (/7H TARGET,I5,20H BRAKE RELEASE TIMES/(16H AIRCRAFT TYPE =
     1,I5,10H STARTS AT,F7.2,9H MINUTES.))
1110  FORMAT (/19H DISTANCE TO TARGET,I4,13H FROM SUB LOCATION,I4,3H IS,
     1F10.2,30H NM. FLIGHT TIME (MISSILE TYPE,I3,3H) =,E16.8,5H MIN.)
1120  FORMAT (7H WEAPON,I5,18H ARRIVES ON TARGET,I5,12H WHEN FIRST ,11HA
     1IRCRAFT IS,F10.2,32H NAUTICAL MILES BEYOND CENTROID.)
1130  FORMAT (13H ANNULUS PK =,F7.4,15H, CIRCULAR PK =,F7.4,9H,  WEAPON,
     1I5,18H VS. AIRCRAFT TYPE,I5,20H.  ANNULAR RADII ARE,F10.4,4H AND,F
     210.4)
1140  FORMAT (36H SURVIVABILITY MATRIX, AIRCRAFT TYPE,I5//)
1150  FORMAT (/7H TARGET,I5)
1160  FORMAT (1X,16F7.4)
1170  FORMAT (F10.4,2I5,F10.4)
1180  FORMAT (/46H LAGRANGE MULTIPLIERS MUST DIFFER BY AT LEAST ,F13.10,
     120H TO CAUSE ITERATION./)
1190  FORMAT (/24H EXPECTED VALUE KILLED =,F10.4)
1200  FORMAT (33H RESTART INFORMATION WRITTEN OUT.)
1210  FORMAT (/48H MULTIPLIER MATRIX CONVERGED WITHIN TOLERANCE OF,F13.1
     10)
1220  FORMAT (/18H MULTIPLIER MATRIX//)
1230  FORMAT (//17H ITERATION NUMBER,I10)
1240  FORMAT (/18H DELTA N IN COLUMN,I4,9H FROM ROW,I4,7H TO ROW,I4,3H I
     1S,F10.3)
1250  FORMAT (/21H KILL CHANGED BY ONLY,F7.4,11H (LESS THAN,F7.4,9H) IN
     1PAST,I5,19H ITERATIONS.  QUIT.)
1260  FORMAT (/42H ITERATION CUT-OFF LIMIT HAS BEEN REACHED.)
1270  FORMAT (/24H CURRENT DELTA LAMBDA IS,F13.10)
1280  FORMAT (33H SURVIVABILITY PRODUCT FOR TARGET,I5,15H, AIRCRAFT TYPE
     1,I5,2H =,F10.4)
1290  FORMAT (1X,F10.4,17H AIRCRAFT OF TYPE,I5,18H MOVED FROM TARGET,I5,
     110H TO TARGET,I5)
1300  FORMAT (27H THIS IS VALUE SHIFT NUMBER,I5)
1310  FORMAT (23H ALLOCATION INTEGERIZED)
1320  FORMAT (20H BEDDOWN INTEGERIZED)
1330  FORMAT (///4H EOJ//)
1340  FORMAT (20H THIS IS MOVE NUMBER,I5,13H OF A SUB.)
1350  FORMAT (/17H SUB POINT NUMBER,5X,14HNUMBER OF SUBS,5X,8HSUB TYPE/(
     1I10,I19,I17))
      END
      SUBROUTINE ALOUT (ITER,NTGTS,NSUBS,NWPNS,ALOC,ISUBS,NMPS,MXTGT)
      DIMENSION ALOC(MXTGT,1), ISUBS(1), NMPS(1)
      CALL PAGE (0)
      CALL PAGE (2)
      PRINT 40, ITER
      DO 10 I=1,NTGTS
      CALL PAGE (-3)
      PRINT 50, I
      J=0
      DO 10 ISUB=1,NSUBS
```

```
      ISLIM=NMPS(ISUB)
      DO 10 ISALVO=1,ISLIM
      J=J+1
      ALOCN=ALOC(I,J)
      IF (ALOCN.LT..0001) GO TO 10
      PRINT 60, ALOCN,ISUBS(ISUB),ISUB,ISALVO,J
      CALL PAGE (1)
10    CONTINUE
      MWPN=0
      DO 35 I=1,NSUBS
      CALL PAGE (-3)
      PRINT 70, I
      JSLIM=NMPS(I)
      IF(ISUBS(I).NE.0)GO TO 15
      MWPN=MWPN+JSLIM
      GO TO 35
   15 DO 30 J=1,JSLIM
      MWPN=MWPN+1
      DO 20 K=1,NTGTS
      ALOCN=ALOC(K,MWPN)
      IF (ALOCN.LT..0001) GO TO 20
      PRINT 80, ALOCN,J,K
      CALL PAGE (1)
20    CONTINUE
30    CONTINUE
35    CONTINUE
      RETURN
C
40    FORMAT (12H ALLOCATION,,5X,16HITERATION NUMBER,I10//)
50    FORMAT (/7H TARGET,I5)
60    FORMAT (3X,F9.4,14H MISSILES FROM,I4,21H SUBS AT SUB LOCATION,I4,7
     1H, SALVO,I4,2X,7H(WEAPON,I5,2H).)
70    FORMAT (/13H SUB LOCATION,I5)
80    FORMAT (3X,F9.4,20H MISSILES FROM SALVO,I4,10H TO TARGET,I4)
      END
      SUBROUTINE ALINT (NTGTS,NSUBS,ALOC,ISUBS,NMPS,MXTGT)
      DIMENSION ALOC(MXTGT,1), ISUBS(1), NMPS(1), ALHOLD(35)
C     INTEGERIZING THE ALLOCATION MATRIX, COLUMN BY COLUMN.
      JWPN=0
      DO 80 J=1,NSUBS
      MLIM=NMPS(J)
      IF (ISUBS(J).NE.0) GO TO 10
      JWPN=JWPN+MLIM
      GO TO 80
10    DO 70 M=1,MLIM
      JWPN=JWPN+1
      DO 20 I=1,NTGTS
20    ALHOLD(I)=0.
      SUMW=0.
      DO 30 I=1,NTGTS
      SUMW=SUMW+ALOC(I,JWPN)
```

```
30    CONTINUE
      NMSLS=SUMH+.01
      IF (NMSLS.EQ.0) GO TO 70
      DO 50 JSUB=1,NMSLS
      HOLD=0.
      DO 40 I=1,NTGTS
      TEST=ALOC(I,JWPN)
      IF (TEST.LE.HOLD) GO TO 40
      HOLD=TEST
      IMAX=I
40    CONTINUE
      ALHOLD(IMAX)=ALHOLD(IMAX)+1.
      ALOC(IMAX,JWPN)=ALOC(IMAX,JWPN)-1.
      IF (ALOC(IMAX,JWPN).LT.0.) ALOC(IMAX,JWPN)=0.
50    CONTINUE
      DO 60 I=1,NTGTS
60    ALOC(I,JWPN)=ALHOLD(I)
70    CONTINUE
80    CONTINUE
      RETURN
      END
      SUBROUTINE VINT (NTGTS,NTYPES,VAL,RELVAL,MXTGT)
      DIMENSION VAL(MXTGT,1), RELVAL(1), VHOLD(35)
      DO 60 JTYPE=1,NTYPES
      RV=RELVAL(JTYPE)
      DO 10 I=1,NTGTS
10    VHOLD(I)=0.
      SUMV=0.
      DO 20 I=1,NTGTS
20    SUMV=SUMV+VAL(I,JTYPE)/RV
      NVAL=SUMV+.01
      DO 40 IVAL=1,NVAL
      HOLD=0.
      DO 30 I=1,NTGTS
      TEST=VAL(I,JTYPE)
      IF (TEST.LE.HOLD) GO TO 30
      HOLD=TEST
      IMAX=I
30    CONTINUE
      VHOLD(IMAX)=VHOLD(IMAX)+RV
      VAL(IMAX,JTYPE)=VAL(IMAX,JTYPE)-RV
      IF (VAL(IMAX,JTYPE).LT.0.) VAL(IMAX,JTYPE)=0.
40    CONTINUE
      DO 50 I=1,NTGTS
50    VAL(I,JTYPE)=VHOLD(I)
60    CONTINUE
      RETURN
      END
      SUBROUTINE TGTKIL (NTGTS,NWPNS,NTYPES,OBJSUM,ALOC,FLAMB,FLAMBI,SUR
     1V,VAL,PROD,VKILL,NWALOC,RELVAL,MXTGT,MXWPNS)
      DIMENSION ALOC(MXTGT,1), FLAMB(MXTGT,1), SURV(MXTGT,MXWPNS,1), FLAT
```

```
     1MBI(MXTGT,MXWPNS,1), VAL(MXTGT,1), PROD(MXTGT,1), VKILL(MXTGT,1),
     2NWALOC(1), RELVAL(1), SUMAC(5), SUMACK(5)
C     COMPUTE FROM SCRATCH, THE PRODUCTS OF EACH TARGET*S SURVIVABILITIE
C     AND VALUE.  COUNT WEAPONS ALLOCATED (BY ROUNDING OFF).
      DO 20 I=1,NTGTS
      NWALOC(I)=0.
      DO 10 JTYPE=1,NTYPES
10    PROD(I,JTYPE)=1.0
      DO 20 J=1,NWPNS
      ALOCN=ALOC(I,J)
      IPART=ALOCN+.5
      NWALOC(I)=NWALOC(I)+IPART
      DO 20 JTYPE=1,NTYPES
      SURVN=SURV(I,J,JTYPE)
20    PROD(I,JTYPE)=PROD(I,JTYPE)*SURVN**ALOCN
C     COMPUTE FROM SCRATCH, THE VALUE KILLED ON EACH TARGET AND THE
C     MULTIPLIER MATRIX.
      OBJSUM=0.
      DO 50 I=1,NTGTS
      DO 30 JTYPE=1,NTYPES
      VKILL(I,JTYPE)=VAL(I,JTYPE)*(1-PROD(I,JTYPE))
30    OBJSUM=OBJSUM+VKILL(I,JTYPE)
      DO 50 J=1,NWPNS
      SUM=0.
      DO 40 JTYPE=1,NTYPES
      ADDEND=-PROD(I,JTYPE)*ALOG(SURV(I,J,JTYPE))*VAL(I,JTYPE)
      FLAMBI(I,J,JTYPE)=ADDEND
40    SUM=SUM+ADDEND
50    FLAMB(I,J)=SUM
C
      ENTRY KILOUT
C
      DO 60 JTYPE=1,NTYPES
      SUMAC(JTYPE)=0.
60    SUMACK(JTYPE)=0.
      CALL PAGE (0)
      CALL PAGE (3)
      PRINT 80
      DO 70 I=1,NTGTS
      CALL PAGE (-NTYPES-1)
      PRINT 90, I,NWALOC(I)
      DO 70 JTYPE=1,NTYPES
      V=VAL(I,JTYPE)
      VK=VKILL(I,JTYPE)
      RV=RELVAL(JTYPE)
      VN=V/RV
      VKN=VK/RV
      PRINT 100, JTYPE,V,VK,VN,VKN
      SUMAC(JTYPE)=SUMAC(JTYPE)+VN
70    SUMACK(JTYPE)=SUMACK(JTYPE)+VKN
      PRINT 110, (JTYPE,SUMAC(JTYPE),SUMACK(JTYPE),JTYPE=1,NTYPES)
```

```
      CALL PAGE (NTYPES+1)
      RETURN
C
80    FORMAT (10X,6HTARGET,9X,7HWEAPONS,8X,8HAIRCRAFT,2(11X,5HTOTAL,10X,
     16HKILLED)/10X,6HNUMBER,7X,9HALLOCATED,12X,4HTYPE,2(11X,5HVALUE),2(
     28X,8HAIRCRAFT)/)
90    FORMAT (2I16)
100   FORMAT (32X,I16,4F16.4)
110   FORMAT (7H TOTALS/(32X,I16,32X,2F16.4))
      END
      SUBROUTINE ADJLAM (NTGTS,NSUBS,NTYPES,NCOL,IBROUT,LOWPT,MAXPT,JCOL
     1,DELTA,EPS,ALOC,FLAMB,FLAMBI,SURV,NWALOC,ISUBS,NMPS,MXTGT,MXWPNS)
      DIMENSION ALOC(MXTGT,1), FLAMB(MXTGT,1), SURV(MXTGT,MXWPNS,1), NWA
     1LOC(1), ISUBS(1), NMPS(1), FLAMBI(MXTGT,MXWPNS,1)
      DATA J,ISUB,ISALVO/0,1,0/
      DO 60 JCNT=1,NCOL
      J=J+1
      ISALVO=ISALVO+1
      IF (ISALVO.LE.NMPS(ISUB)) GO TO 20
      ISALVO=1
10    ISUB=ISUB+1
      IF (ISUB.LE.NSUBS) GO TO 20
      ISUB=1
      J=1
20    IF (ISUBS(ISUB).NE.0) GO TO 30
      J=J+NMPS(ISUB)
      GO TO 10
30    JCOL=J
      FMAX=-1.
      MAXPT=1
      FMIN=9999.
      DO 50 I=1,NTGTS
      FTESTL=FLAMB(I,J)
      IF (FTESTL.LE.FMAX) GO TO 40
      FMAX=FTESTL
      MAXPT=I
40    IF (ALOC(I,J).LT..0001.OR.FTESTL.GE.FMIN) GO TO 50
      FMIN=FTESTL
      LOWPT=I
50    CONTINUE
      IF (FMIN.LT.FMAX-EPS) GO TO 70
60    CONTINUE
C     NO EXCHANGES OF ALLOCATION POSSIBLE.
      IBROUT=1
      RETURN
C     COMPUTE INCREMENT IN ALLOCATION.
70    ALOW=ALOC(LOWPT,JCOL)
      IF (NTYPES.NE.1) GO TO 100
      IF (FMIN/FMAX.LT..0001) GO TO 80
      DELTA=ALOG(FMIN/FMAX)/ALOG(SURV(LOWPT,JCOL,1)*SURV(MAXPT,JCOL,1))
      GO TO 90
```

```
80    DELTA=9999.
90    IF (DELTA.GT.ALOW) DELTA=ALOW
      GO TO 110
100   DELTA=XNEWT(0.,ALOW,EPS,LOWPT,MAXPT,JCOL,NTYPES,SURV,FLAMBI,MXTGT,
     1MXWPNS)
110   AMAX=ALOC(MAXPT,JCOL)
      IPART1=AMAX+.5
      ANEW=AMAX+DELTA
      IPART=ANEW+.5
      NWALOC(MAXPT)=NWALOC(MAXPT)-IPART1+IPART
      ALOC(MAXPT,JCOL)=ANEW
      IPART1=ALOW+.5
      ANEW=ALOW-DELTA
      IPART=ANEW+.5
      NWALOC(LOWPT)=NWALOC(LOWPT)-IPART1+IPART
      ALOC(LOWPT,JCOL)=ANEW
      RETURN
      END
      FUNCTION XNEWT (XMIN,XMAX,EPS,LOWPT,MAXPT,JCOL,NTYPES,SURV,FLAMBI,
     1MXTGT,MXWPNS)
      DIMENSION SURV(MXTGT,MXWPNS,1), FLAMBI(MXTGT,MXWPNS,1)
      DATA XBND/200./
      CALL FOFP (XMAX,LOWPT,MAXPT,JCOL,NTYPES,FMAX,DIV,SURV,FLAMBI,MXTGT
     1,MXWPNS)
      IF (FMAX.LT.0.0) GO TO 60
      IST=0
      XNEW=XMAX-DIV
10    IF (ABS(XNEW).LT.XBND) GO TO 50
      IST=IST+1
      GO TO (20,30,40), IST
20    CALL FOFP (XMIN,LOWPT,MAXPT,JCOL,NTYPES,DUMMY,DIV,SURV,FLAMBI,MXTG
     1T,MXWPNS)
      XNEW=XMIN-DIV
      GO TO 10
30    XMID=(XMAX+XMIN)/2.
      CALL FOFP (XMID,LOWPT,MAXPT,JCOL,NTYPES,DUMMY,DIV,SURV,FLAMBI,MXTG
     1T,MXWPNS)
      XNEW=XMID-DIV
      GO TO 10
40    PRINT 70, IST
      XNEWT=XMID
      RETURN
50    XOLD=XNEW
      CALL FOFP (XOLD,LOWPT,MAXPT,JCOL,NTYPES,DUMMY,DIV,SURV,FLAMBI,MXTG
     1T,MXWPNS)
      XNEW=XOLD-DIV
      IF (ABS(XNEW-XOLD).GT.EPS) GO TO 10
      XNEWT=XNEW
      RETURN
60    XNEWT=XMAX
      RETURN
```

```
C
70    FORMAT (16H NEWTON DIVERGED,I5,27H TIMES.  XMID USED AS ROOT.)
      END
      SUBROUTINE FOFP (X,LOWPT,MAXPT,JCOL,NTYPES,SUM,DIV,SURV,FLAMBI,MXT
     1GT,MXWPNS)
      DIMENSION SURV(MXTGT,MXWPNS,1), FLAMBI(MXTGT,MXWPNS,1)
      SUM=0.
      PRIME=0.
      DO 10 JTYPE=1,NTYPES
      FLLOW=FLAMBI(LOWPT,JCOL,JTYPE)
      FLHI=FLAMBI(MAXPT,JCOL,JTYPE)
      SLOW=SURV(LOWPT,JCOL,JTYPE)
      SHI=SURV(MAXPT,JCOL,JTYPE)
      PLOW=FLLOW*SLOW**(-X)
      PHI=FLHI*SHI**X
      SUM=SUM+PLOW-PHI
10    PRIME=PRIME-PLOW*ALOG(SLOW)-PHI*ALOG(SHI)
      DIV=SUM/PRIME
      RETURN
      END
      SUBROUTINE SUBADJ (NTGTS,NSUBS,MTYPES,EPS,IBOUT,ALOC,FLAMB,ISUBS,N
     1MPS,MTYPE,MXTGT)
      DIMENSION LOHOLD(16), MXHOLD(16), LOTEMP(16), MXTEMP(16), LOH(16),
     1 MXH(16), ALOC(MXTGT,1), FLAMB(MXTGT,1), ISUBS(1), NMPS(1), MTYPE(
     21)
      BIGDIF=-1.
      CALL PAGE (0)
      DO 110 JTYPE=1,MTYPES
      HOLDLO=1.0E+300
      HOLDHI=-1.
      JWPN=0
      DO 90 J=1,NSUBS
      NMIS=NMPS(J)
      IF (MTYPE(J).NE.JTYPE) GO TO 80
      IBRSW=1
      IF (ISUBS(J).EQ.0) IBRSW=2
      SUMLO=0.
      SUMHI=0.
      NM=NMIS
      DO 40 M=1,NMIS
      JWPN=JWPN+1
      HLO=1.0E+300
      HHI=-1.
      DO 30 I=1,NTGTS
      TEST=FLAMB(I,JWPN)
      IF (TEST.LE.HHI) GO TO 10
      HHI=TEST
      MXTEMP(M)=I
10    GO TO (20,30), IBRSW
20    IF (TEST.GE.HLO.OR.ALOC(I,JWPN).LT..0001) GO TO 30
      HLO=TEST
```

```
      LOTEMP(M)=I
30    CONTINUE
      SUMLO=SUMLO+HLO
40    SUMHI=SUMHI+HHI
      IF (SUMHI.LE.HOLDHI) GO TO 60
      HOLDHI=SUMHI
      MAXS=J
      MAXWS=JWPN-NMIS
      DO 50 M=1,NMIS
50    MXH(M)=MXTEMP(M)
60    IF (SUMLO.GE.HOLDLO) GO TO 90
      HOLDLO=SUMLO
      LOWS=J
      LOWWS=JWPN-NMIS
      DO 70 M=1,NMIS
70    LOH(M)=LOTEMP(M)
      GO TO 90
80    JWPN=JWPN+NMIS
90    CONTINUE
      CALL PAGE (-3)
      PRINT 180, JTYPE,HOLDLO,HOLDHI
      AVDIF=(HOLDHI-HOLDLO)/NM
      IF (AVDIF.LE.BIGDIF) GO TO 110
      BIGDIF=AVDIF
      MAXSUB=MAXS
      MAXW=MAXWS
      LOWSUB=LOWS
      LOWW=LOWWS
      DO 100 M=1,NM
      MXHOLD(M)=MXH(M)
100   LOHOLD(M)=LOH(M)
110   CONTINUE
      IF (BIGDIF.GE.EPS) GO TO 120
      IBOUT=1
      PRINT 190
      RETURN
C     FIND MAX LAMBDA IN COLUMN OF WEAPON TO BE ADDED.
120   MLIM=NMPS(MAXSUB)
      DO 170 M=1,MLIM
      MAXWPN=MAXW+M
      LAMAX=MXHOLD(M)
C     ADD A WEAPON IN APPROPRIATE SPOT.
      ALOC(LAMAX,MAXWPN)=ALOC(LAMAX,MAXWPN)+1.0
      LOWWPN=LOWW+M
      LAMLOW=LOHOLD(M)
C     MOVE WEAPON FRACTIONS UNTIL ONE WEAPON MOVED.
      ALFRAC=0.
      GO TO 150
130   FMIN=1.0E+300
      DO 140 I=1,NTGTS
      TEST=FLAMB(I,LOWWPN)
```

```
      IF (TEST.GE.FMIN.OR.ALOC(I,LOWWPN).LT..001) GO TO 140
      FMIN=TEST
      LAMLOW=I
140   CONTINUE
150   AL=ALOC(LAMLOW,LOWWPN)
      IF (AL.GT..999-ALFRAC) GO TO 160
      ALFRAC=ALFRAC+AL
      ALOC(LAMLOW,LOWWPN)=0.
      GO TO 130
160   ALOC(LAMLOW,LOWWPN)=ALOC(LAMLOW,LOWWPN)-(1.0-ALFRAC)
170   CONTINUE
      ISUBS(MAXSUB)=ISUBS(MAXSUB)+1
      ISUBS(LOWSUB)=ISUBS(LOWSUB)-1
      PRINT 200, LOWSUB,MAXSUB
      CALL PAGE (1)
      RETURN
C
180   FORMAT (/17H FOR MISSILE TYPE,I5/28H SUM OF LOWEST LAMBDAS WITH ,9
     1HWEAPONS =,F10.4/25H SUM OF HIGHEST LAMBDAS =,F10.4)
190   FORMAT (/15H SUBS CONVERGED)
200   FORMAT (24H SUB MOVED FROM LOCATION,I5,12H TO LOCATION,I5)
      END
      FUNCTION PROFLU (XLU,X,Y,NPTS)
      DIMENSION X(NPTS), Y(NPTS)
      XHI=X(NPTS)
      XLO=X(1)
      YLO=Y(1)
      SLOPE=(Y(NPTS)-Y(NPTS-1))/(XHI-X(NPTS-1))
      IF (XLU.LE.XHI.AND.XLU.GE.XLO) GO TO 10
      IF (XLU.GT.XHI) PROFLU=Y(NPTS)+SLOPE*(XLU-XHI)
      IF (XLU.LT.XLO) PROFLU=YLO
      RETURN
10    PROFLU=ALAG(XLU,X,Y,NPTS)
      IF (PROFLU.LE.YLO) PROFLU=YLO
      RETURN
      END
      FUNCTION XAREA (XL,XD,AL,AH)
      DIMENSION NUMBER(17), MUMBER(8)
      COMMON /LIST1/ PI
      DATA (NUMBER(J),J=1,17)/2121,2221,2321,2331,3121,3212,3221,3222,33
     112,3321,3322,3331,3332,4221,4321,4331,5321/
      DATA (MUMBER(J),J=1,8)/512,611,612,711,712,722,812,912/
C
C
C     XAREA SUBROUTINE RETURNS THE INTERSECTION OF THE CIRCULAR LETHAL
C     AREA OF A WEAPON WITH AN ANNULUS FORMED BY THE MAXIMUM AND
C     MINIMUM AIRCRAFT FLYOUT DISTANCES FROM A CENTROID.
C
C
C     INPUT VARIABLE DESCRIPTION-
C
```

```
C        XL -- THE LETHAL RADIUS OF THE KILL CIRCLE OF THE WEAPON
C        XD -- THE HORIZONTAL DISPLACEMENT OF THE POINT OF
C              DETONATION OF THE WEAPON FROM THE CENTROID
C        AL -- THE MINIMUM AIRCRAFT FLYOUT DISTANCE FROM THE CENTROID
C        AH -- THE MAXIMUM AIRCRAFT FLYOUT DISTANCE FROM THE CENTROID
C
C
C     RESTRICTIONS-
C
C        1.  ALL NUMBERS MUST BE REAL AND NON-NEGATIVE.
C        2.  AH MUST BE GREATER THEN AL.
C        3.  ALL NUMBERS MUST HAVE THE SAME UNITS.
C
C
C     ********************************************************
C
C           ROUTINE DEVELOPED BY BILL PEAY, SAB, X2295
C                          12 OCTOBER 1972
C
C     ********************************************************
C
C
      PI=3.1415926536
      XMIN=-1000000000.0
      XMAX=1000000000.0
      IF (AL.GE.AH) GO TO 20
      IF (XL.GT.AH) GO TO 190
      A=XD-XL
      IF (A.GE.(-AH).AND.A.LE.(-AL)) I4=2000
      IF (A.GT.(-AL).AND.A.LE.(AL)) I4=3000
      IF (A.GT.AL.AND.A.LE.AH) I4=4000
      IF (A.GT.AH) I4=5000
      B=XD+XL
      IF (B.LE.AL) I3=100
      IF (B.GT.AL.AND.B.LE.AH) I3=200
      IF (B.GT.AH) I3=300
      IF (I4.EQ.3000.AND.I3.GE.200) XMIN=(XD**2+AL**2-XL**2)/(2.0*XD)
      IF (I3.EQ.300.AND.I4.LE.4000) XMAX=(XD**2+AH**2-XL**2)/(2.0*XD)
      IF (XD.LE.XMIN) I2=10
      IF (XD.GT.XMIN.AND.XD.LE.XMAX) I2=20
      IF (XD.GT.XMAX) I2=30
      IF (XMIN.LE.0.0) I1=1
      IF (XMIN.GT.0.0) I1=2
      NUM=I4+I3+I2+I1
      ITST=1
      DO 10 J=1,17
      IF (NUM.GE.NUMBER(J)) ITST=ITST+1
10    CONTINUE
      GO TO (20,30,100,150,160,30,40,110,50,120,170,130,180,140,60,70,80
     1,90), ITST
20    XAREA=-1.0
```

```
      RETURN
30    XAREA=0.0
      RETURN
40    XAREA=SUB02(XL,XD,AL,AH)
      RETURN
50    XAREA=SUB03(XL,XD,AL,AH)
      RETURN
60    XAREA=PI*XL**2
      RETURN
70    XAREA=SUB05(XL,XD,AL,AH)
      RETURN
80    XAREA=SUB06(XL,XD,AL,AH)
      RETURN
90    XAREA=0.0
      RETURN
100   XAREA=PI*(XL**2-AL**2)
      RETURN
110   XAREA=SUB09(XL,XD,AL,AH)
      RETURN
120   XAREA=SUB11(XL,XD,AL,AH)
      RETURN
130   XAREA=SUB12(XL,XD,AL,AH)
      RETURN
140   XAREA=SUB13(XL,XD,AL,AH)
      RETURN
150   XAREA=SUB05(XL,XD,AL,AH)-PI*AL**2
      RETURN
160   XAREA=SUB06(XL,XD,AL,AH)-PI*AL**2
      RETURN
170   XAREA=SUB16(XL,XD,AL,AH)
      RETURN
180   XAREA=SUB17(XL,XD,AL,AH)
      RETURN
190   A=XD-XL
      IF (A.LE.(-AH)) I4=500
      IF (A.GT.(-AH).AND.A.LE.(-AL)) I4=600
      IF (A.GT.(-AL).AND.A.LE.(AL)) I4=700
      IF (A.GT.AL.AND.A.LE.AH) I4=800
      IF (A.GT.AH) I4=900
      IF (I4.EQ.700) XMIN=(XD**2+XL**2-AL**2)/(2.0*XD)
      IF (I4.GE.600.AND.I4.LE.800) XMAX=(XD**2+XL**2-AH**2)/(2.0*XD)
      IF (XMIN.LE.0.0) I3=10
      IF (XMIN.GT.0.0) I3=20
      IF (XMAX.LE.0.0) I2=1
      IF (XMAX.GT.0.0) I2=2
      NUM=I4+I3+I2
      ITST=1
      DO 200 J=1,8
      IF (NUM.GE.NUMBER(J)) ITST=ITST+1
200   CONTINUE
      GO TO (210,220,230,240,250,260,270,80,90), ITST
```

```
210   XAREA=-1.0
      RETURN
220   XAREA=PI*(AH**2-AL**2)
      RETURN
230   XAREA=SUB19(XL,XD,AL,AH)
      RETURN
240   XAREA=SUB06(XL,XD,AL,AH)-PI*AL**2
      RETURN
250   XAREA=SUB21(XL,XD,AL,AH)
      RETURN
260   XAREA=SUB22(XL,XD,AL,AH)
      RETURN
270   XAREA=SUB23(XL,XD,AL,AH)
      RETURN
      END
      FUNCTION SUB02 (XL,XD,AL,AH)
      ALANG=(ACOS((AL**2+XD**2-XL**2)/(2.0*AL*XD)))*2.0
      OPANG=ACOS((AL**2+XL**2-XD**2)/(2.0*AL*XL))
      XLANG=(ALANG/2.0+OPANG)*2.0
      ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
      BSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0
      SUB02=ASEGMT-BSEGMT
      RETURN
      END
      FUNCTION SUB03 (XL,XD,AL,AH)
      COMMON /LIST1/ PI
      ALANG=(ACOS((AL**2+XD**2-XL**2)/(2.0*AL*XD)))*2.0
      XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0
      ASEGMT=(XL**2)*(PI-(XLANG-SIN(XLANG))/2.0)
      BSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0
      SUB03=ASEGMT-BSEGMT
      RETURN
      END
      FUNCTION SUB05 (XL,XD,AL,AH)
      COMMON /LIST1/ PI
      AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0
      OPANG=ACOS((AH**2+XL**2-XD**2)/(2.0*AH*XL))
      XLANG=(AHANG/2.0+OPANG)*2.0
      ASEGMT=(XL**2)*(PI-(XLANG-SIN(XLANG))/2.0)
      BSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0
      SUB05=ASEGMT+BSEGMT
      RETURN
      END
      FUNCTION SUB06 (XL,XD,AL,AH)
      AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0
      XLANG=(ACOS((XL**2+XD**2-AH**2)/(2.0*XL*XD)))*2.0
      ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
      BSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0
      SUB06=ASEGMT+BSEGMT
      RETURN
      END
```

```
FUNCTION SUB09 (XL,XD,AL,AH)
COMMON /LIST1/ PI
XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0
OPANG=ACOS((XL**2+AL**2-XD**2)/(2.0*XL*AL))
ALANG=(XLANG/2.0+OPANG)*2.0
ASEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0
BSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
SUB09=PI*(XL**2-AL**2)+ASEGMT-BSEGMT
RETURN
END
FUNCTION SUB11 (XL,XD,AL,AH)
ALANG=(ACOS((AL**2+XD**2-XL**2)/(2.0*AL*XD)))*2.0
OPANG=ACOS((AL**2+XL**2-XD**2)/(2.0*AL*XL))
XLANG=(ALANG/2.0+OPANG)*2.0
ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
BSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0
AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0
OPANG=ACOS((AH**2+XL**2-XD**2)/(2.0*AH*XL))
XLANG=(AHANG/2.0+OPANG)*2.0
CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
DSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0
SUB11=ASEGMT+DSEGMT-BSEGMT-CSEGMT
RETURN
END
FUNCTION SUB12 (XL,XD,AL,AH)
COMMON /LIST1/ PI
ALANG=(ACOS((AL**2+XD**2-XL**2)/(2.0*AL*XD)))*2.0
XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0
ASEGMT=(XL**2)*(PI-(XLANG-SIN(XLANG))/2.0)
BSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0
AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0
OPANG=ACOS((AH**2+XL**2-XD**2)/(2.0*AH*XL))
XLANG=(AHANG/2.0+OPANG)*2.0
CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
DSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0
SUB12=ASEGMT+DSEGMT-BSEGMT-CSEGMT
RETURN
END
FUNCTION SUB13 (XL,XD,AL,AH)
ALANG=(ACOS((AL**2+XD**2-XL**2)/(2.0*AL*XD)))*2.0
XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0
ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
BSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0
AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0
XLANG=(ACOS((XL**2+XD**2-AH**2)/(2.0*XL*XD)))*2.0
CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
DSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0
SUB13=CSEGMT+DSEGMT-ASEGMT-BSEGMT
RETURN
END
FUNCTION SUB16 (XL,XD,AL,AH)
```

```
      COMMON /LIST1/ PI
      XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0
      OPANG=ACOS((XL**2+AL**2-XD**2)/(2.0*XL*AL))
      ALANG=(XLANG/2.0+OPANG)*2.0
      ASEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0
      BSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
      AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0
      OPANG=ACOS((AH**2+XL**2-XD**2)/(2.0*AH*XL))
      XLANG=(AHANG/2.0+OPANG)*2.0
      CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
      DSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0
      SUB16=PI*(XL**2-AL**2)+ASEGMT+DSEGMT-BSEGMT-CSEGMT
      RETURN
      END
      FUNCTION SUB17 (XL,XD,AL,AH)
      COMMON /LIST1/ PI
      XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0
      OPANG=ACOS((XL**2+AL**2-XD**2)/(2.0*XL*AL))
      ALANG=(XLANG/2.0+OPANG)*2.0
      ASEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0
      BSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
      AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0
      XLANG=(ACOS((XL**2+XD**2-AH**2)/(2.0*XL*XD)))*2.0
      CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
      DSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0
      SUB17=ASEGMT+CSEGMT+DSEGMT-BSEGMT-PI*AL**2
      RETURN
      END
      FUNCTION SUB19 (XL,XD,AL,AH)
      COMMON /LIST1/ PI
      AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0
      OPANG=ACOS((AH**2+XL**2-XD**2)/(2.0*AH*XL))
      XLANG=(AHANG/2.0+OPANG)*2.0
      ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
      BSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0
      SUB19=PI*(AH**2-AL**2)-ASEGMT+BSEGMT
      RETURN
      END
      FUNCTION SUB21 (XL,XD,AL,AH)
      COMMON /LIST1/ PI
      AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0
      OPANG=ACOS((AH**2+XL**2-XD**2)/(2.0*AH*XL))
      XLANG=(AHANG/2.0+OPANG)*2.0
      ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
      BSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0
      XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0
      OPANG=ACOS((XL**2+AL**2-XD**2)/(2.0*XL*AL))
      ALANG=(XLANG/2.0+OPANG)*2.0
      CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
      DSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0
      SUB21=PI*(AH**2-AL**2)+ASEGMT+DSEGMT-BSEGMT-CSEGMT
```

```
      RETURN
      END
      FUNCTION SUB22 (XL,XD,AL,AH)
      COMMON /LIST1/ PI
      AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0
      XLANG=(ACOS((XL**2+XD**2-AH**2)/(2.0*XL*XD)))*2.0
      ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
      BSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0
      XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0
      OPANG=ACOS((XL**2+AL**2-XD**2)/(2.0*XL*AL))
      ALANG=(XLANG/2.0+OPANG)*2.0
      CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
      DSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0
      SUB22=ASEGMT+BSEGMT+DSEGMT-CSEGMT-PI*AL**2
      RETURN
      END
      FUNCTION SUB23 (XL,XD,AL,AH)
      AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0
      XLANG=(ACOS((XL**2+XD**2-AH**2)/(2.0*XL*XD)))*2.0
      ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
      BSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0
      XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0
      ALANG=(ACOS((AL**2+XD**2-XL**2)/(2.0*AL*XD)))*2.0
      CSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0
      DSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0
      SUB23=ASEGMT+BSEGMT-CSEGMT-DSEGMT
      RETURN
      END
      FUNCTION DIST (XLAT,XLNG,YLAT,YLNG)
C     THIS FUNCTION COMPUTES THE GREAT CIRCLE DISTANCE (IN N.M.) BETWEEN
C     POINTS X AND Y WITH LATITUDES AND LONGITUDES GIVEN IN TERMS OF
C     RADIANS (NORTH AND WEST ARE POSITIVE).
      DATA PI2/1.57079632/
      A=ABS(XLNG-YLNG)
      B=PI2-XLAT
      C=PI2-YLAT
      DIST=3442.2*ACOS(COS(B)*COS(C)+SIN(B)*SIN(C)*COS(A))
      RETURN
      END
      SUBROUTINE PAGE (N)
C
      DATA LINE/1/
C
10    IF (N) 20,50,90
20    IF (LINE-N-50) 30,40,40
30    LINE=LINE-N
      RETURN
C
40    LINE=-N
      GO TO 70
C
```

```
50    IF (LINE) 60,80,60                                                PGE 130
60    LINE=0                                                            PGE 140
70    PRINT 100                                                         PGE 150
80    RETURN                                                            PGE 160
C                                                                       PGE 170
90    LINE=LINE+N                                                       PGE 180
      IF (LINE-50) 80,60,60                                             PGE 190
C                                                                       PGE 200
C                                                                       PGE 210
100   FORMAT (1H1)                                                      PGE 220
      END                                                               PGE 230-
      SUBROUTINE PROCESS (PSIN,ICALN,YIELDM,NCASE,MODE)                 PRO  10
      DIMENSION PSIN(1), ICALN(1), YIELDM(1)                            PRO  20
      COMMON /DISTIME/ S(99,2),D(99,2),NUMDATA(2),CV(2),TI(2),ITYPE,NTYPPRO  30
     1E,JTYPE,MTYPE,RADMAX                                              PRO  40
      COMMON /ACFTS/ A(50,2),F(50,2),G(50,2),VS(50,2),VEL(50,2),ACCEL(50PRO  50
     1,2),NDATA(2)                                                      PRO  60
      COMMON /NUCLER/ FLRP(2,4),FTSA(2,4),FLRT(2,4),BURST(2),DISMIN(2),VPRO  70
     1PSI(2),VCAL(2),VYIELD(4)                                          PRO  80
C                                                                       PRO  90
C     THIS SUBROUTINE IS USED TO INPUT DATA FOR AIRCRAFT AND IS USED    PRO 100
C     TO GENERATE NUCLEAR EFFECTS GEOMETRY FOR SPECIFIED HARDNESS       PRO 110
C     LEVELS VIA SUBROUTINES SABERCM AND SNAPTCM                        PRO 120
C                                                                       PRO 130
      COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBLPRO 140
     1,VLL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,DBURST,MBURST,ATM,CAL,TEFFI   PRO 150
      COMMON /PROBLEM/ NEWPROB                                          PRO 160
      LOGICAL NEWPROB                                                   PRO 170
      DATA TOL,PERTB1,PERTB2,CMTF,CFTM/0.001,0.001,1.0001,3.2808,0.3048/PRO 180
      DO 210 I=1,NTYPE                                                  PRO 190
      CALL PAGE (0)                                                     PRO 200
      WRITE (6,220) NCASE,MODE                                          PRO 210
      READ 310, AZ,HBL                                                  PRO 220
      BURST(I)=HBL                                                      PRO 230
      AZORIG=AZ                                                         PRO 240
      READ 230, N                                                       PRO 250
      NDATA(I)=N                                                        PRO 260
      WRITE (6,240) I,NTYPE                                             PRO 270
      CAL=FLOAT(ICALN(I))                                               PRO 280
      DELP=PSIN(I)                                                      PRO 290
      VPSI(I)=DELP                                                      PRO 300
      VCAL(I)=CAL                                                       PRO 310
      WRITE (6,250) I                                                   PRO 320
      WRITE (6,260) DELP,ICALN(I)                                       PRO 330
      WRITE (6,270) I                                                   PRO 340
      WRITE (6,280)                                                     PRO 350
      CALL PAGE (23)                                                    PRO 360
      MPTS=0                                                            PRO 380
C                                                                       PRO 390
C     THE FOLLOWING SECTION READS AIRCRAFT INPUT DATA AND ADJUSTS THE   PRO 400
C     ALTITUDE (IF NECESSARY) FOR USE BY ALAG WHICH REQUIRES IT TO BE   PRO 410
```

```
C     STRICTLY MONOTONICALLY INCREASING. THE ORIGINAL ALTITUDE INPUT
C     DATA IS ASSUMED TO BE MONOTONICALLY INCREASING.
C     AN AUTOMATIC LOOKUP ON THE VELOCITY OF SOUND GIVEN ALTITUDE IS
C     PERFORMED VIA FUNCTION SUBROUTINE SSPF.
C
      READ 290, F(1,I),G(1,I),A(1,I),VEL(1,I),ACCEL(1,I)
      ALT=CFTM*A(1,I)
      VS(1,I)=CMTF*SSPF(ALT)
      J=1
      WRITE (6,300) J,F(1,I),G(1,I),A(1,I),VS(1,I),VEL(1,I),ACCEL(1,I)
      DO 40 J=2,N
      READ 290, F(J,I),G(J,I),A(J,I),VEL(J,I),ACCEL(J,I)
10    IF (A(J,I).GT.A(J-1,I)) GO TO 30
      A(J,I)=PERTB2*A(J-1,I)
20    IF (AZ.EQ.A(J-1,I)) AZ=A(J,I)
      IF (A(J,I).GT.0.0) GO TO 10
      A(J,I)=A(J-1,I)+PERTB1
      GO TO 20
30    ALT=CFTM*A(J,I)
      VS(J,I)=CMTF*SSPF(ALT)
      WRITE (6,300) J,F(J,I),G(J,I),A(J,I),VS(J,I),VEL(J,I),ACCEL(J,I)
      CALL PAGE (1)
40    CONTINUE
      READ 290, FALTCM,FMACH,TI(I),XATI
      READ 310, HTE,BETIND,RHO,VIS,PZ,HSL
      FALT=AZ
      VS1=ALAG(AZ,A(1,I),VS(1,I),N)
      VEL1=ALAG(AZ,A(1,I),VEL(1,I),N)
      IF ((ABS(VEL1-FMACH)/((VEL1+FMACH)/2.0)).LE.0.01) VEL1=FMACH
      VOE=VS1*VEL1
      ACC=ALAG(AZ,A(1,I),ACCEL(1,I),N)
      CALL PAGE (-5)
      WRITE (6,320) FALTCM
      CALL PAGE (0)
      WRITE (6,330) I,AZ
      WRITE (6,340) VS1
      WRITE (6,350) ACC
      IF ((VEL1.LT.FMACH).AND.(XATI.EQ.0.0)) WRITE (6,360) VEL1,VEL1
      IF ((VEL1.GE.FMACH).OR.((VEL1.LT.FMACH).AND.(XATI.GT.0.0))) WRITE
     1(6,360) VEL1,FMACH
      VF=VS1*FMACH
      IF ((VEL1.LT.FMACH).AND.(XATI.EQ.0.0)) WRITE (6,370) VOE,VOE
      IF ((VEL1.GE.FMACH).OR.((VEL1.LT.FMACH).AND.(XATI.GT.0.0))) WRITE
     1(6,370) VOE,VF
      IF (VEL1.LE.FMACH) GO TO 50
      WRITE (6,380)
      STOP
50    CONTINUE
C
C
C     THE VARIABLE DISMIN CONTAINS THE MINIMUM DISTANCE FROM BRAKE
```

```
C       RELEASE AT WHICH THE LEVEL-OFF ALTITUDE IS ATTAINED.
C
        DISMIN(I)=ALAG(AZORIG,A(1,I),F(1,I),N)
C
C
C       EQUATIONS OF MOTION
C
        ALT=FALT
        V0=V0E
        L=NDATA(I)
        ALTCM=FALTCM
        ACCI02=ACC/2.0
        XJUMP=0.0
        IF (VEL1.EQ.FMACH) XJUMP=1.0
        IF ((VEL1.LT.FMACH).AND.(XATI.EQ.0.0)) XJUMP=1.0
        IF (ALT.GE.ALTCM) XJUMP=1.0
        S0=ALAG(ALT,A(1,I),F(1,I),L)
        T0=ALAG(ALT,A(1,I),G(1,I),L)
        IF (XJUMP.EQ.0.0) GO TO 60
        CV(I)=V0
        GO TO 80
60      T0SQ=T0*T0
        C1=ACC*T0
        VSOUND=ALAG(ALT,A(1,I),VS(1,I),L)
        CV(I)=FMACH*VSOUND
        TF=T0+((CV(I)-V0)/ACC)
        IF (TF.GT.T0) GO TO 70
        WRITE (6,390) T0,TF,CV(I),V0
        STOP
70      DT=(TF-T0)/XATI
80      DO 90 J=1,L
        K=J
        NUMDATA(I)=K
        IF ((F(J,I).GE.S0).OR.(G(J,I).GE.T0)) GO TO 100
        D(J,I)=F(J,I)
        S(J,I)=G(J,I)
90      CONTINUE
100     D(K,I)=S0
        S(K,I)=T0
        IF (XJUMP.NE.0.0) GO TO 140
110     K=K+1
        S(K,I)=S(K-1,I)+DT
        IF (S(K,I).GE.TF) GO TO 120
        D(K,I)=S0+(V0*(S(K,I)-T0))+(ACCI02*(S(K,I)**2+T0SQ))-(C1*S(K,I))
        GO TO 110
120     S(K,I)=TF
        D(K,I)=S0+(V0*(S(K,I)-T0))+(ACCI02*(S(K,I)**2+T0SQ))-(C1*S(K,I))
        IF (ABS(S(K,I)-S(K-1,I)).LT.TOL) GO TO 130
        NUMDATA(I)=K
        GO TO 140
130     NUMDATA(I)=K-1
```

```
140   NUMDATA(I)=NUMDATA(I)+1
      INDEX=NUMDATA(I)
      S(INDEX,I)=S(INDEX-1,I)+TI(I)
      D(INDEX,I)=(CV(I)*TI(I))+D(INDEX-1,I)
      IF (NUMDATA(I).LT.80) GO TO 140
C
      WRITE (6,400)
      CALL PAGE (27)
C     FOLLOWING CONVERTS FEET/SECONDS TO NM/MINUTES
      INDEX=NUMDATA(I)
      DO 150 KK=1,INDEX
      TEMP1=S(KK,I)
      TEMP2=D(KK,I)
      S(KK,I)=S(KK,I)/60.0
      D(KK,I)=D(KK,I)/6080.0
      WRITE (6,410) TEMP2,TEMP1,D(KK,I),S(KK,I)
      CALL PAGE (1)
150   CONTINUE
      CV(I)=CV(I)*60.0/6080.0
      TI(I)=TI(I)/60.0
      DISMIN(I)=DISMIN(I)/6080.0
C
      DO 200 J=1,MTYPE
      W=YIELDM(J)
      VYIELD(J)=W
      WRITE (6,420) J,MTYPE,I
      WRITE (6,430) J,W
      READ 440, ISABER,HORF,TSA
      WRITE (6,450)
      WRITE (6,460)
      WRITE (6,470) DELP
      WRITE (6,480) W
      WRITE (6,490) HTE
      WRITE (6,500) HBL
      WRITE (6,510) AZ
      WRITE (6,520)
      IF (ISABER.NE.1) GO TO 160
      WRITE (6,530)
      GO TO 170
160   IPROB1=0
      HORF=TSA=0.0
      CALL SABERCM (SR,AZ,HTE,HBL,W,DELP,HORF,TSA,NCASE,IPROB1)
      IF (IPROB1.NE.1) GO TO 170
      WRITE (6,540)
      HORF=TSA=0.0
170   FLRP(I,J)=HORF/6080.0
      FTSA(I,J)=TSA/60.0
      WRITE (6,550) FLRP(I,J),HORF,FTSA(I,J),TSA
      READ 440, ISNAPT,SZ
      WRITE (6,560)
      WRITE (6,570)
```

```
      WRITE (6,580) CAL
      WRITE (6,590) W
      WRITE (6,600) HTE
      WRITE (6,610) HBL
      WRITE (6,620) AZ
      WRITE (6,630) HSL
      WRITE (6,640) PZ
      WRITE (6,650) VIS
      WRITE (6,660) RHO
      WRITE (6,670) BETIND
      WRITE (6,680)
      IF (ISNAPT.NE.1) GO TO 180
      WRITE (6,690)
      GO TO 190
180   IPROB2=0
      SZ=0.0
      CALL SNAPTCM (IPROB2)
      IF ((IPROB2.NE.1).AND.(.NOT.NEWPROB)) GO TO 190
      IF (IPROB2.EQ.1) WRITE (6,700)
      IF (NEWPROB) WRITE (6,710)
      SZ=0.0
190   FLRT(I,J)=SZ/6080.0
      WRITE (6,720) FLRT(I,J),SZ
200   CONTINUE
210   CONTINUE
      RETURN
C
220   FORMAT (28H SUBROUTINE PROCESS,    CASE ,I3,9H,    MODE ,I2////)
230   FORMAT (I5)
240   FORMAT (1H ,5X,14HAIRCRAFT TYPE ,I2,5H  OF ,I2,9H  TYPE(S),///)
250   FORMAT (1H ,10X,41HVULNERABILITY CRITERIA FOR AIRCRAFT TYPE ,I2//)
260   FORMAT (1H ,15X,F5.2,5H  PSI/16X,I5,8H  CAL/CM,3H**2///)
270   FORMAT (1H ,10X,29HDATA INPUT FOR AIRCRAFT TYPE ,I2//)
280   FORMAT (1H ,16X,4HCARD,10X,6HGROUND,15X,6HFLIGHT,14X,8HAIRCRAFT,13
     1X,8HVELOCITY,11X,4HMACH,5X,9HLEVEL-OFF/16X,6HNUMBER,9X,5HRANGE,17X
     2,4HTIME,15X,8HALTITUDE,13X,8HOF SOUND,10X,6HNUMBER,3X,12HACCELERAT
     3ION/)
290   FORMAT (3F15.8,2F10.6)
300   FORMAT (1H ,16X,I3,4(5X,E16.8),2(6X,F6.3))
310   FORMAT (6F10.8)
320   FORMAT (2(/,1X),15X,46HINITIAL ALTITUDE (IN CLIMBING) OF MAXIMUM M
     1ACH//21X,F7.0,6H  FEET)
330   FORMAT (11X,32HDATA COMPUTED FOR AIRCRAFT TYPE ,I2,15H  WITH RESPE
     1CT ,26HTO A TERMINAL ALTITUDE OF ,F7.0,6H  FEET//)
340   FORMAT (1H ,15X,17HVELOCITY OF SOUND//21X,F8.2,13H  FEET/SECOND//)
350   FORMAT (1H ,15X,22HACCELERATION COMPONENT//21X,F6.3,20H  FEET/SECO
     1ND/SECOND//)
360   FORMAT (1H ,15X,12HMACH NUMBERS//21X,9H INITIAL ,F6.3/21X,9HTERMIN
     1AL ,F6.3//)
370   FORMAT (1H ,15X,8HVELOCITY//21X,9H INITIAL ,F7.1,13H  FEET/SECOND/
     121X,9HTERMINAL ,F7.1,13H  FEET/SECOND//)
```

```
380   FORMAT (77H STOP ISSUED - INITIAL VELOCITY (VEL1) GREATER THAN TERPRO2480
     1MINAL VELOCITY (FMACH))                                          PRO2490
390   FORMAT (1H1,4HTG= ,E16.8,10X,4HTF= ,E16.8,10X,3HCV=,E16.8,10X,3HVGPRO2500
     1=,E16.8//35H  DATA INPUT INCORRECT---(CV.LE.VG))                 PRO2510
400   FORMAT (17X,39HGROUND RANGE (FEET)   FLIGHT TIME (SEC),18X,37HGROUPRO2520
     1ND RANGE (NM)   FLIGHT TIME (MIN)/)                              PRO2530
410   FORMAT (20X,E16.8,4X,E16.8,19X,E16.8,4X,E16.8)                   PRO2540
420   FORMAT (1H1,11X,13HMISSILE TYPE ,I2,5H  OF ,I2,32H  TYPE(S) AGAINSPRO2550
     1T AIRCRAFT TYPE ,I2//)                                           PRO2560
430   FORMAT (1H ,15X,22HYIELD OF MISSILE TYPE ,I2//20X,F6.0,4H  KT//)  PRO2570
440   FORMAT (I5,2F15.8)                                               PRO2580
450   FORMAT (14(/,1X),14X,18HSUBROUTINE SABERCM/)                     PRO2590
460   FORMAT (1H ,20X,21HDATA INPUT TO SABERCM)                        PRO2600
470   FORMAT (1H ,26X,F5.2,26H  PSI      BLAST OVERPRESSURE)           PRO2610
480   FORMAT (1H ,25X,F6.0,15H  KT       YIELD)                        PRO2620
490   FORMAT (1H ,25X,F6.0,24H  FEET     TERRAIN HEIGHT)               PRO2630
500   FORMAT (1H ,25X,F6.0,22H  FEET     BURST HEIGHT)                 PRO2640
510   FORMAT (1H ,25X,F6.0,27H  FEET     AIRCRAFT ALTITUDE/)           PRO2650
520   FORMAT (1H ,15X,22HSABERCM OUTPUT FOLLOWS)                       PRO2660
530   FORMAT (39H  ***** SABERCM OVERRIDE EFFECTED *****,11H ISABER = 1)PRO2670
540   FORMAT (39H  ***** SABERCM PROBLEMS DETECTED *****,11H IPROB1 = 1)PRO2680
550   FORMAT (1H-,18X,91H  A SUMMARY OF DATA OUTPUT FROM SABERCM USED BYPRO2690
     1 SUBROUTINE DETAREA IN COMPUTING LETHAL AREA/26X,29HLETHAL OVERPREPRO2700
     2SSURE RADIUS = ,E16.8,16H  NAUTICAL MILES/53X,2H= ,E16.8,6H  FEET/PRO2710
     331X,24HTIME OF SHOCK ARRIVAL = ,E16.8,9H  MINUTES/53X,2H= ,E16.8,9PRO2720
     4H  SECONDS//)                                                    PRO2730
560   FORMAT (1H1,15X,18HSUBROUTINE SNAPTCM/)                          PRO2740
570   FORMAT (1H ,20X,21HDATA INPUT TO SNAPTCM)                        PRO2750
580   FORMAT (1H ,26X,F5.0,8H  CAL/CM,3H**2,16H  THERMAL ENERGY)       PRO2760
590   FORMAT (1H ,25X,F6.0,18H  KT           YIELD)                    PRO2770
600   FORMAT (1H ,25X,F6.0,27H  FEET         TERRAIN HEIGHT)           PRO2780
610   FORMAT (1H ,25X,F6.0,25H  FEET         BURST HEIGHT)             PRO2790
620   FORMAT (1H ,25X,F6.0,30H  FEET         AIRCRAFT ALTITUDE)        PRO2800
630   FORMAT (1H ,25X,F6.0,30H  FEET         HAZE LAYER HEIGHT)        PRO2810
640   FORMAT (1H ,25X,F6.1,33H  MM HG        WATER VAPOR PRESSURE)     PRO2820
650   FORMAT (1H ,25X,F6.1,44H  MILES        VISIBILITY(U.S., STATUTE MILEPRO2830
     1S))                                                              PRO2840
660   FORMAT (1H ,25X,F6.2,31H  (ALBEDO)     GROUND REFLECTANCE)       PRO2850
670   FORMAT (1H ,25X,F6.2,26H  (BETINO)     SHOULD BE 1.0/)           PRO2860
680   FORMAT (1H ,15X,22HSNAPTCM OUTPUT FOLLOWS)                       PRO2870
690   FORMAT (39H  ***** SNAPTCM OVERRIDE EFFECTED *****,11H ISNAPT = 1)PRO2880
700   FORMAT (39H  ***** SNAPTCH PROBLEMS DETECTED *****,11H IPROB2 = 1)PRO2890
710   FORMAT (39H  ***** SNAPTCH PROBLEMS DETECTED *****,17HNEWPROB = .TPRO2900
     1RU E.)                                                           PRO2910
720   FORMAT (1H-,18X,91H  A SUMMARY OF DATA OUTPUT FROM SNAPTCH USED BYPRO2920
     1 SUBROUTINE DETAREA IN COMPUTING LETHAL AREA/26X,24HLETHAL THERMALPRO2930
     2 RADIUS = ,E16.8,16H  NAUTICAL MILES/48X,2H= ,E16.8,6H  FEET//)   PRO2940
      END                                                              PRO2950-
      FUNCTION SSPF (HM)                                               SSP  10
C                                                                      SSP  20
C     ROUTINE TO COMPUTE SOUND SPEED IN METERS/SECOND AS A FUNCTION OF  SSP  30
```

```
C      GEOMETRIC ALTITUDE IN METERS.
C      BASED ON DATA PRESENTED IN //U.S. STANDARD ATMOSPHERE, 1962//.
C
C      ROUTINE BY HARRY H. MURPHY, JR.,  29APR71, CORRECTED 3OCT72 (HMM)
C
       DIMENSION ALT(10), TM(10), DTDZ(10)
C
       DATA ALT/-4996.0,0.0,11019.0,20063.0,32162.0,47350.0,52429.0,61591
     1.0,79994.0,90000.0/
C
       DATA TM/320.65,288.15,216.65,216.65,228.65,270.65,270.65,252.65,18
     10.65,180.65/
C
       DATA DTDZ/-6.5052E-3,-6.4888E-3,0.0,0.91826-4,2.7653E-3,0.0,-1.997
     14E-3,-3.9124E-3,0.0,0.0/
C
C
10     Z=HM
       I=1
       IF (Z+5000.0) 50,50,20
20     IF (90000.0-Z) 60,60,30
C
30     DO 40 I=1,9
       IF (ALT(I+1)-Z) 40,40,50
40     CONTINUE
       I=10
C
50     SSPF=20.046796*SQRT(TM(I)+DTDZ(I)*(Z-ALT(I)))
       RETURN
C
60     SSPF=269.44
       RETURN
       END
       FUNCTION DETAREA (Q,DSPT)
       DIMENSION X1(200), Y1(200), X2(200), Y2(200), X(400), Y(400)
       COMMON /DISTIME/ S(99,2),D(99,2),NUMDATA(2),CV(2),TI(2),ITYPE,NTYP
     1E,JTYPE,MTYPE,RADMAX
       COMMON /ACFTS/ A(50,2),F(50,2),G(50,2),VS(50,2),VEL(50,2),ACCEL(50
     1,2),NDATA(2)
       COMMON /NUCLER/ FLRP(2,4),FTSA(2,4),FLRT(2,4),BURST(2),DISMIN(2),V
     1PSI(2),VCAL(2),VYIELD(4)
       COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBL
     1,VLL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,JBURST,MBURST,ATM,CAL,TEFFI
       COMMON /PROBLEM/ NEWPROB
       LOGICAL NEWPROB
       DATA PI/3.141592653589793/
C
C      ******************************************************************
C      THE LATTER PART OF THIS SUBROUTINE COMPUTES LETHAL AREA FOR A
C      BURST LOCATED Q NAUTICAL MILES FROM THE CENTROID
C      ******************************************************************
```

```
C
      IF (Q.LT.0.0) STOP
      QDNM=Q+DSPT
      IF (QDNM.GE.DISMIN(ITYPE)) GO TO 30
C
      QDFT=6080.0*QDNM
      AZ=ALAG(QDFT,F(1,ITYPE),A(1,ITYPE),NDATA(ITYPE))
      IF (AZ.LT.HTE) AZ=HTE
      HBL=BURST(ITYPE)
      WRITE (6,140) Q,DSPT,QDNM,DISMIN(ITYPE)
      CAL=VCAL(ITYPE)
      DELP=VPSI(ITYPE)
      W=VYIELD(JTYPE)
C
      WRITE (6,150) JTYPE,MTYPE,ITYPE
      WRITE (6,160) JTYPE,W
      WRITE (6,170)
      WRITE (6,180)
      WRITE (6,190) DELP
      WRITE (6,200) W
      WRITE (6,210) HTE
      WRITE (6,220) HBL
      WRITE (6,230) AZ
      WRITE (6,240)
      IPROB1=0
      HORF=TSA=0.0
      CALL SABERCM (SR,AZ,HTE,HBL,W,DELP,HORF,TSA,NCASE,IPROB1)
      IF (IPROB1.NE.1) GO TO 10
      WRITE (6,250)
      HORF=TSA=0.0
10    FLRPSI=HORF/6080.0
      FTSASI=TSA/60.0
      WRITE (6,260) FLRPSI,HORF,FTSASI,TSA
      WRITE (6,270)
      WRITE (6,280)
      WRITE (6,290) CAL
      WRITE (6,300) W
      WRITE (6,310) HTE
      WRITE (6,320) HBL
      WRITE (6,330) AZ
      WRITE (6,340) HSL
      WRITE (6,350) PZ
      WRITE (6,360) VIS
      WRITE (6,370) RHO
      WRITE (6,380) BETIND
      WRITE (6,390)
      IPROB2=0
      SZ=0.0
      CALL SNAPTCM (IPROB2)
      IF ((IPROB2.NE.1).AND.(.NOT.NEWPROB)) GO TO 20
      IF (IPROB2.EQ.1) WRITE (6,400)
```

```
      IF (NEWPROB) WRITE (6,410)
      SZ=0.0
20    FLRTHM=SZ/6080.0
      WRITE (6,420) FLRTHM,SZ
C
      R=HORF/6080.0
      T=TSA/60.0
      RTHERM2=SZ/6080.0
      GO TO 40
C
C
30    R=FLRP(ITYPE,JTYPE)
      RTHERM2=FLRT(ITYPE,JTYPE)
      T=FTSA(ITYPE,JTYPE)
40    IF ((Q.EQ.0.0).OR.(R.LE.0.0).OR.(T.LE.0.0)) GO TO 110
      INDEX=NUMDATA(ITYPE)
      TSPT=ALAG(DSPT,D(1,ITYPE),S(1,ITYPE),INDEX)
      IF (Q.GT.R) GO TO 90
C
C     AT THIS POINT THE VARIABLES R,T,RTHERM2,DSPT, AND TSPT ARE KNOWN.
C     ALSO IT HAS BEEN ESTABLISHED THAT (Q.GT.0.0),(R.GT.0.0),(T.GT.0.0)
C     AND (Q.LE.R)
C     NEED TO COMPUTE ACIR, THE DISTANCE FROM THE CENTROID THAT THE
C     AIRCRAFT CAN TRAVEL IN TIME T, ASSUMING THAT THE AIRCRAFT WAS
C     LOCATED AT THE CENTROID AT THE ONSET OF THE BURST.
C
      T2=TSPT+T
      ACIR=ALAG(T2,S(1,ITYPE),D(1,ITYPE),INDEX)-DSPT
      IF (ACIR.GT.0.0) GO TO 50
      WRITE (6,430) ACIR
      IF (ACIR.LT.0.0) STOP
50    ACIRSQ=ACIR*ACIR
      IF ((Q+R).LE.ACIR) GO TO 120
C
C
      IF (Q.LT.R) GO TO 70
C
C
C     AT THIS POINT Q.EQ.R
      IF (ACIR.EQ.0.0) GO TO 60
C
C     AT THIS POINT ACIR.GT.0.0
C     ALSO ACIR.LT.(Q+R)=(2.0*R)
      XAR=ACIRSQ/(2.0*R)
      YAR=SQRT(ACIRSQ-(XAR*XAR))
      ALPHA=ATAN(YAR/XAR)
      GO TO 100
C
C
60    ALPHA=PI/2.0
      GO TO 100
```

```
C
C
C     AT THIS POINT Q.LT.R
70    IF ((R-Q).GE.ACIR) GO TO 80
C
C
C     AT THIS POINT  (R-Q).LT.ACIR.LT.(Q+R)
      XAR=((Q*Q)-(R*R)+ACIRSQ)/(2.0*Q)
      YAR=SQRT(ACIRSQ-(XAR*XAR))
      IF (XAR.GT.0.0) ALPHA=ATAN(YAR/XAR)
      IF (XAR.EQ.0.0) ALPHA=PI/2.0
      IF (XAR.LT.0.0) ALPHA=PI-ATAN(YAR/(-XAR))
      GO TO 100
C
C
C     AT THIS POINT  (R-Q).GE.ACIR
80    ALPHA=PI
      GO TO 100
C
C
C     AT THIS POINT Q.GT.R
90    ALPHA=ASIN(R/Q)
100   BETA=0.0
      RADMAX=RTHERM2
      CALL EXTRACT (X1,Y1,X2,Y2,IALPHA,R,Q,TSPT,X,Y,NPTS,T,ALPHA,RTHERM2
     1,BETA)
      IF (BETA.EQ.0.0) SECTOR=0.0
      IF (BETA.GT.0.0) SECTOR=BETA*RTHERM2*RTHERM2
      IF (BETA.LT.0.0) STOP
      IF (NPTS.EQ.0) GO TO 120
      IF (NPTS.LT.0) RADMAX=X(-NPTS)-Q
      IF (NPTS.LT.0) NPTS=-NPTS
      DETAREA=TRITSYM(X,Y,NPTS)+SECTOR
      RETURN
C
110   IF (R.GT.RTHERM2) GO TO 130
120   RADMAX=RTHERM2
      DETAREA=PI*RTHERM2*RTHERM2
      RETURN
C
C     AT THIS POINT  (R.GT.RTHERM2)
130   INDEX=NUMDATA(ITYPE)
      TSPT=ALAG(DSPT,D(1,ITYPE),S(1,ITYPE),INDEX)
      RBOP=BAKUP(R,T,TSPT)
      RMAX=AMAX1(RTHERM2,RBOP)
      RADMAX=RMAX
      DETAREA=PI*RMAX*RMAX
      RETURN
C
140   FORMAT (1H1,48HSUBROUTINE DETAREA NUCLEAR LOOKUP - (Q+DSPT) = (,F6
     1.2,3H + ,F6.2,4H) = ,F6.2,20H NM, WHERE DISMIN = ,F6.2,3H NM/)
```

```
150   FORMAT (1H ,11X,13HMISSILE TYPE ,I2,5H  OF ,I2,32H  TYPE(S) AGAINSDET1720
     1T AIRCRAFT TYPE ,I2/)                                             DET1730
160   FORMAT (1H ,15X,22HYIELD OF MISSILE TYPE ,I2/,20X,F6.0,4H  KT/)  DET1740
170   FORMAT (14(/,1X),14X,18HSUBROUTINE SABERCM/)                      DET1750
180   FORMAT (1H ,20X,21HDATA INPUT TO SABERCM)                         DET1760
190   FORMAT (1H ,26X,F5.2,26H  PSI      BLAST OVERPRESSURE)            DET1770
200   FORMAT (1H ,25X,F6.0,15H  KT       YIELD)                         DET1780
210   FORMAT (1H ,25X,F6.0,24H  FEET     TERRAIN HEIGHT)                DET1790
220   FORMAT (1H ,25X,F6.0,22H  FEET     BURST HEIGHT)                  DET1800
230   FORMAT (1H ,25X,F6.0,27H  FEET     AIRCRAFT ALTITUDE/)            DET1810
240   FORMAT (1H ,15X,22HSABERCM OUTPUT FOLLOWS)                        DET1820
250   FORMAT (39H  ***** SABERCM PROBLEMS DETECTED *****,11H IPROB1 = 1)DET1830
260   FORMAT (1H-,18X,91H  A SUMMARY OF DATA OUTPUT FROM SABERCM USED BYDET1840
     1 SUBROUTINE DETAREA IN COMPUTING LETHAL AREA/26X,29HLETHAL OVERPREDET1850
     2SSURE RADIUS = ,E16.8,16H  NAUTICAL MILES/53X,2H= ,E16.8,6H  FEET/DET1860
     331X,24HTIME OF SHOCK ARRIVAL = ,E16.8,9H  MINUTES/53X,2H= ,E16.8,9DET1870
     4H  SECONDS//)                                                     DET1880
270   FORMAT (1H1,15X,18HSUBROUTINE SNAPTCM/)                           DET1890
280   FORMAT (1H ,20X,21HDATA INPUT TO SNAPTCM)                         DET1900
290   FORMAT (1H ,26X,F5.0,8H  CAL/CM,3H**2,16H  THERMAL ENERGY)        DET1910
300   FORMAT (1H ,25X,F6.0,18H  KT          YIELD)                      DET1920
310   FORMAT (1H ,25X,F6.0,27H  FEET        TERRAIN HEIGHT)             DET1930
320   FORMAT (1H ,25X,F6.0,25H  FEET        BURST HEIGHT)               DET1940
330   FORMAT (1H ,25X,F6.0,30H  FEET        AIRCRAFT ALTITUDE)          DET1950
340   FORMAT (1H ,25X,F6.0,30H  FEET        HAZE LAYER HEIGHT)          DET1960
350   FORMAT (1H ,25X,F6.1,33H  MM HG       WATER VAPOR PRESSURE)       DET1970
360   FORMAT (1H ,25X,F6.1,44H  MILES       VISIBILITY(U.S., STATUTE MILEDET1980
     1S))                                                               DET1990
370   FORMAT (1H ,25X,F6.2,31H  (ALBEDO)    GROUND REFLECTANCE)         DET2000
380   FORMAT (1H ,25X,F6.2,26H  (BETINO)    SHOULD BE 1.0/)             DET2010
390   FORMAT (1H ,15X,22HSNAPTCM OUTPUT FOLLOWS)                        DET2020
400   FORMAT (39H  ***** SNAPTCM PROBLEMS DETECTED *****,11H IPROB2 = 1)DET2030
410   FORMAT (39H  ***** SNAPTCM PROBLEMS DETECTED *****,17HNEWPROB = .TDET2040
     1RU E.)                                                            DET2050
420   FORMAT (1H-,18X,91H  A SUMMARY OF DATA OUTPUT FROM SNAPTCM USED BYDET2060
     1 SUBROUTINE DETAREA IN COMPUTING LETHAL AREA/26X,24HLETHAL THERMALDET2070
     2 RADIUS = ,E16.8,16H  NAUTICAL MILES/48X,2H= ,E16.8,6H  FEET//)   DET2080
430   FORMAT (1H1,130(1H*),//,7H ACIR =,E16.8,//,1X,130(1H*))           DET2090
      END                                                               DET2100-
      FUNCTION TRITSYM (X,Y,N)                                          TRI   10
      DIMENSION X(1), Y(1)                                              TRI   20
      TRITSYM=0.0                                                       TRI   30
      DO 10 J=2,N                                                       TRI   40
      TRITSYM=TRITSYM+(X(J-1)*Y(J))-(X(J)*Y(J-1))                       TRI   50
10    CONTINUE                                                          TRI   60
      TRITSYM=ABS(TRITSYM)                                              TRI   70
      RETURN                                                            TRI   80
      END                                                               TRI   90-
      SUBROUTINE EXTRACT (X1,Y1,X2,Y2,IALPHA,R,Q,TSPT,X,Y,NPTS,T,ALPHA,REXT   10
     1THERM2,BETA)                                                      EXT   20
      DIMENSION RADSQ(4)                                                EXT   30
```

```
      DIMENSION X(1), Y(1), X1(1), Y1(1), X2(1), Y2(1)
      DATA PI/3.141592653589793/
      I=0
      K=0
      RSQ=R**2
      QSQ=Q**2
      THETA=0.0
      DELTA=ALPHA/150.0
      RTHERSQ=RTHERM2*RTHERM2
      IF (Q.EQ.R) GO TO 190
      IF (Q.LT.R) GO TO 220
10    I=I+1
      STHET=SIN(THETA)
      CTHET=COS(THETA)
      C1=Q*CTHET
      STSQ=STHET**2
      C1A=RSQ-QSQ*STSQ
      IF (C1A.LT.0.0) GO TO 30
      C2=SQRT(C1A)
      R1=C1+C2
      R2=C1-C2
20    IF (T.GT.0.0) R1C=BAKUP(R1,T,TSPT)
      IF (T.GT.0.0) R2C=BAKUP(R2,T,TSPT)
      IF (T.EQ.0.0) R1C=R1
      IF (T.EQ.0.0) R2C=R2
      X2(I)=R1C*CTHET
      Y2(I)=R1C*STHET
      X1(I)=R2C*CTHET
      Y1(I)=R2C*STHET
      IF (K.EQ.1) GO TO 40
      THETA=THETA+DELTA
      IF (THETA.LT.ALPHA) GO TO 10
      I=I+1
30    K=1
      STHET=SIN(ALPHA)
      CTHET=COS(ALPHA)
      R1=Q*CTHET
      R2=R1
      GO TO 20
40    DO 50 J=1,I
      X(J)=X1(J)
      Y(J)=Y1(J)
      X(I+J)=X2(I-J+1)
      Y(I+J)=Y2(I-J+1)
50    CONTINUE
      M=2*I
60    DO 70 J=1,M
      IF ((((X(J)-Q)**2)+(Y(J)**2)).GE.RTHERSQ) GO TO 70
      NPTS=J-1
      IF (NPTS.GE.1) GO TO 100
C     AT THIS POINT (NPTS.EQ.0)
```

```
C     THERMAL CIRCLE CONTAINS PETAL LOCUS - RETURN TO DETAREA WITH
C     NPTS.EQ.0 AND USE THERMAL RADIUS
      RETURN
70    CONTINUE
C
C     AT THIS POINT THERE IS NOT A SINGLE POINT OF THE OVERPRESSURE
C     LOCUS WHICH LIES EITHER ON OR INSIDE OF THE THERMAL CIRCLE.
C     TWO POSSIBILITIES EXIST,
C        (1) THE OVERPRESSURE LOCUS CONTAINS THE THERMAL CIRCLE
C        (2) THE OVERPRESSURE LOCUS DOES NOT CONTAIN, RATHER LIES
C            OUTSIDE OF THE THERMAL CIRCLE
C
C     IF X(M).LE.(Q-RTHERM2) THEN (2) HOLDS
C     IF X(M).GE.(Q+RTHERM2) THEN (1) HOLDS
C
      IF (X(M).GE.(Q+RTHERM2)) GO TO 90
C
C     AT THIS POINT (2) HOLDS
80    BETA=PI
      NPTS=M
      WRITE (6,390)
      RETURN
C
C
90    BETA=0.0
      WRITE (6,400)
      NPTS=-M
      RETURN
C
100   NR=4
C
C     AT THIS POINT THE OVERPRESSURE LOCUS INTERSECTS THE THERMAL CIRCLE
C     THE POINT GIVEN BY X(NPTS),Y(NPTS) LIES OUTSIDE OF THE THERMAL
C     CIRCLE AND THE POINT GIVEN BY X(NPTS+1),Y(NPTS+1) LIES INSIDE OF
C     THE THERMAL CIRCLE. THE REMAINING TASK IS TO FIND BETA.
C     THE CONDITION HOLDS THAT (1.LE.NPTS.LE.(M-1))
C     MUST ESTABLISH WHETHER OR NOT ((NPTS+3).GT.M)
C
      IF ((NPTS+3).GT.M) GO TO 140
      RADSQ(4)=((X(NPTS)-Q)**2)+(Y(NPTS)**2)
      RADSQ(3)=((X(NPTS+1)-Q)**2)+(Y(NPTS+1)**2)
      RADSQ(2)=((X(NPTS+2)-Q)**2)+(Y(NPTS+2)**2)
      RADSQ(1)=((X(NPTS+3)-Q)**2)+(Y(NPTS+3)**2)
C
      IF ((RADSQ(1).EQ.RADSQ(2)).AND.(RADSQ(2).EQ.RADSQ(3))) STOP
      IF (RADSQ(1).EQ.RADSQ(2)) GO TO 110
      IF (RADSQ(2).EQ.RADSQ(3)) GO TO 130
C
      MP1=M+1
      X(MP1)=X(NPTS+3)
      Y(MP1)=Y(NPTS+3)
```

```
      X(MP1+1)=X(NPTS+2)
      Y(MP1+1)=Y(NPTS+2)
      X(MP1+2)=X(NPTS+1)
      Y(MP1+2)=Y(NPTS+1)
      X(MP1+3)=X(NPTS)
      Y(MP1+3)=Y(NPTS)
      XINT=ALAG(RTHERSQ,RADSQ,X(MP1),NR)
      YINT=ALAG(RTHERSQ,RADSQ,Y(MP1),NR)
      GO TO 150
C
C     AT THIS POINT RADSQ(1) EQUALS RADSQ(2)
110   X(M+2)=X(NPTS+2)
      Y(M+2)=Y(NPTS+2)
120   X(M+4)=X(NPTS)
      Y(M+4)=Y(NPTS)
      X(M+3)=X(NPTS+1)
      Y(M+3)=Y(NPTS+1)
      X(M+1)=X(NPTS+4)
      Y(M+1)=Y(NPTS+4)
      MP1=M+1
      RADSQ(1)=((X(MP1)-Q)**2)+(Y(MP1)**2)
      XINT=ALAG(RTHERSQ,RADSQ,X(MP1),NR)
      YINT=ALAG(RTHERSQ,RADSQ,Y(MP1),NR)
      GO TO 150
C
C     AT THIS POINT RADSQ(2) EQUALS RADSQ(3)
130   X(M+2)=X(NPTS+3)
      Y(M+2)=Y(NPTS+3)
      RADSQ(2)=RADSQ(1)
      GO TO 120
C
C
140   RADSQ(1)=((X(M)-Q)**2)+(Y(M)**2)
      RADSQ(2)=((X(M-1)-Q)**2)+(Y(M-1)**2)
      RADSQ(3)=((X(M-2)-Q)**2)+(Y(M-2)**2)
      RADSQ(4)=((X(M-3)-Q)**2)+(Y(M-3)**2)
      X(M+1)=X(M-1)
      Y(M+1)=Y(M-1)
      X(M+2)=X(M-2)
      Y(M+2)=Y(M-2)
      X(M+3)=X(M-3)
      Y(M+3)=Y(M-3)
      XINT=ALAG(RTHERSQ,RADSQ,X(M),NR)
      YINT=ALAG(RTHERSQ,RADSQ,Y(M),NR)
C
C     MUST ESTABLISH WHERE THE POINT GIVEN BY XINT,YINT LIES WITH
C     RESPECT TO THE THE POINT GIVEN BY Q,0.0
C
150   IF (I.EQ.1000) GO TO 340
      IF (XINT.LT.Q) GO TO 170
      IF (XINT.GT.Q) GO TO 160
```

```
C
C      WE HAVE (XINT.EQ.Q)
       BETA=PI/2.0
       GO TO 180
C
C      WE HAVE (XINT.GT.Q)
160    BETA=ATAN(YINT/(XINT-Q))
       GO TO 180
C
C      WE HAVE (XINT.LT.Q)
170    BETA=PI-ATAN(YINT/(Q-XINT))
C
C
180    NPTS=NPTS+1
       X(NPTS)=XINT
       Y(NPTS)=YINT
       NPTS=NPTS+1
       X(NPTS)=Q
       Y(NPTS)=0.0
       RETURN
C
C
C      AT THIS POINT Q.EQ.R
C      THE Q.EQ.R ALGORITHM FOLLOWS
C
190    TWOR=2.0*R
200    I=I+1
       STHET=SIN(THETA)
       CTHET=COS(THETA)
       R1=TWOR*CTHET
210    IF (T.GT.0.0) R1C=BAKUP(R1,T,TSPT)
       IF (T.EQ.0.0) R1C=R1
       X1(I)=R1C*CTHET
       Y1(I)=R1C*STHET
C
       IF (K.EQ.1) GO TO 250
       THETA=THETA+DELTA
       IF (THETA.LT.ALPHA) GO TO 200
       I=I+1
       K=1
       STHET=SIN(ALPHA)
       CTHET=COS(ALPHA)
       R1=TWOR*CTHET
       GO TO 210
C
C
C      AT THIS POINT Q.LT.R
C      THE Q.LT.R ALGORITHM FOLLOWS
C
220    I=I+1
       STHET=SIN(THETA)
```

```
      CTHET=COS(THETA)
      C1=Q*CTHET
      C1A=RSQ-(QSQ*STHET*STHET)
      IF (C1A.LT.0.0) GO TO 240
      C2=SQRT(C1A)
      R1=C1+C2
230   IF (T.GT.0.0) R1C=BAKUP(R1,T,TSPT)
      IF (T.EQ.0.0) R1C=R1
      X1(I)=R1C*CTHET
      Y1(I)=R1C*STHET
C
      IF (K.EQ.1) GO TO 250
      THETA=THETA+DELTA
      IF (THETA.LT.ALPHA) GO TO 220
      I=I+1
240   K=1
      STHET=SIN(ALPHA)
      CTHET=COS(ALPHA)
      R1=(Q*CTHET)+SQRT(RSQ-(QSQ*STHET*STHET))
      GO TO 230
C
C
250   K=0
      KM=0
      KP=0
      DO 260 J=1,I
      X(J)=X1(I-J+1)
      Y(J)=Y1(I-J+1)
      DISQ=((X(J)-Q)**2)+(Y(J)**2)
      IF (DISQ.LE.RTHERSQ) X2(J)=-1.0
      IF (DISQ.LE.RTHERSQ) KM=KM+1
      IF (DISQ.GT.RTHERSQ) X2(J)=1.0
      IF (DISQ.GT.RTHERSQ) KP=KP+1
      IF (J.EQ.1) GO TO 260
      IF (X2(J).NE.X2(J-1)) K=K+1
260   CONTINUE
      M=I
C
C
C     THE VARIABLE K CONTAINS THE NUMBER OF TIMES THAT THE OVERPRESSURE
C     LOCUS CROSSES THE THERMAL LOCUS.
C
C     IF  KP = (+I)  THEN ALL POINTS OF THE OVERPRESSURE LOCUS LIE
C                    OUTSIDE OF THE THERMAL CIRCLE (K=0)
C     IF  KM = (+I)  THEN ALL POINTS OF THE OVERPRESSURE LOCUS LIE
C                    EITHER ON OR INSIDE OF THE THERMAL CIRCLE (K=0)
C
C
      IF (RTHERM2.GT.0.0) GO TO 270
C
C     AT THIS POINT RTHERM2=0.0 WHEN ENTERED FROM ABOVE
```

```
      NPTS=-I
      BETA=0.0
      RETURN
C
C
C     AT THIS POINT RTHERM2.GT.0.0
270   IF (K.GT.0) GO TO 300
      IF (KM.NE.I) GO TO 290
C
C     AT THIS POINT  KM=(+I) WHEN ENTERED FROM ABOVE
C     THE OVERPRESSURE LOCUS CONTRIBUTES LITTLE OR NOTHING
280   NPTS=0
      RETURN
C
C
C     AT THIS POINT  KP=(+I) WHEN ENTERED FROM ABOVE
C     THE THERMAL CIRCLE IS CONTAINED BY OR LIES OUTSIDE OF AND DOES NOT
C     CONTAIN THE OVERPRESSURE LOCUS
290   M=I
      IF (X(I).GT.(Q+RTHERM2)) GO TO 90
      GO TO 80
C
C
300   IF (K.GT.1) GO TO 370
C
C     AT THIS POINT THE MOST TRIVIAL CASES INVOLVING NON-INTERSECTION
C     OF THE LETHAL LOCI HAVE BEEN DETECTED AND ISOLATED.
C     THE NEXT MOST SIMPLE CASE IS THAT OF FIRST ORDER INTERSECTION.
C     FIRST ORDER INTERSECTION OCCURS WHEN THE POSITIVE HALF OF THE
C     OVERPRESSURE LOCUS INTERSECTS THE THERMAL CIRCLE ONLY ONCE.
C     THE POSITIVE HALF OF THE OVERPRESSURE LOCUS HAS BEEN GENERATED
C     ABOVE BY THE Q.LT.R OR THE Q.EQ.R ALGORITHM.
C     THE COORDINATES OF THE OVERPRESSURE LOCUS ARE CONTAINED IN THE
C     ARRAYS X AND Y. THE POINT GIVEN BY COORDINATES X(1),Y(1) IS
C     NEAREST THE CENTROID - THE POINT GIVEN BY X(I),Y(I) IS FARTHEST
C     FROM THE CENTROID.
C
C     FIRST ORDER INTERSECTION OCCURS WHEN K=1
C     FIRST ORDER INTERSECTION MAY BE CONSIDERED BY SENSING THE VALUES
C     CONTAINED IN THE ARRAY X2(J) FOR J=1,2,....,I, THE CONTENTS OF
C     THE ARRAY X2(J) SHOULD RESEMBLE (FOR FIRST ORDER INTERSECTION)
C     ONE OF THE FOLLOWING CASES,
C                     CASE(1) +1,+1,....,+1,-1,-1,....,-1
C                     CASE(2) -1,-1,....,-1,+1,+1,....,+1
C                     CASE(3) +1,-1,-1,....,-1
C                     CASE(4) -1,-1,....,-1,+1
C                     CASE(5) +1,+1,....,+1,-1
C                     CASE(6) -1,+1,+1,....,+1
C
C
C     CASES (3) AND (4), ALL OF OVERPRESSURE LOCUS(EXCEPT FOR ONE POINT)
```

```
C     IS EITHER ON OR WITHIN THE THERMAL CIRCLE
      IF (KM.EQ.(I-1)) GO TO 280
C
C
C     CASES (5) AND (6), ALL OF OVERPRESSURE LOCUS(EXCEPT FOR ONE POINT)
C     IS OUTSIDE OF THE THERMAL CIRCLE
      IF (KP.NE.(I-1)) GO TO 310
C
C     THE FOLLOWING IF IS TRUE FOR CASE (6)
      IF ((X2(1).EQ.(-1.0)).AND.(X2(I).EQ.(+1.0))) GO TO 290
C
C     IT IS ESTABLISHED INDIRECTLY THAT WE HAVE CASE (5)
      M=I
      IF (X(I).GE.Q) GO TO 90
      GO TO 80
C
C
C     CASES (1) AND (2) REMAIN TO BE DETECTED FOR K=1
C
C
C     THE FOLLOWING IS TRUE FOR CASE (1)
310   IF ((X2(1).EQ.(+1.0)).AND.(X2(I).EQ.(-1.0))) GO TO 60
C
C
C     AT THIS POINT CASE (2) REMAINS
C     CASE (2) IS ALMOST IDENTICAL TO CASE (1) AFTER INTERCHANGING
C     X(J) WITH X(I-J+1) AND Y(J) WITH Y(I-J+1) FOR J=1,2,...
C     THE COMPUTATION OF BETA IS DIFFERENT THOUGH.
C     VARIABLE I IS SET EQUAL TO 1000 AS A FLAG SO THAT THE BETA
C     COMPUTATION IS PERFORMED AT STATEMENT 390 RATHER THAN AT 90
      M=I
      DO 320 J=1,I
      IF ((I-J+1).LE.J) GO TO 330
      TEMP=X(J)
      X(J)=X(I-J+1)
      X(I-J+1)=TEMP
      TEMP=Y(J)
      Y(J)=Y(I-J+1)
      Y(I-J+1)=TEMP
320   CONTINUE
330   I=1000
      GO TO 60
C
C
340   IF (XINT.LT.Q) GO TO 360
      IF (XINT.GT.Q) GO TO 350
C
C     WE HAVE (XINT.EQ.0.0)
      BETA=PI/2.0
      GO TO 180
C
```

```
C      WE HAVE (XINT.GT.Q)
350    BETA=PI-ATAN(YINT/(XINT-Q))
       GO TO 180
C
C      WE HAVE (XINT.LT.Q)
360    BETA=ATAN(YINT/(Q-XINT))
       GO TO 180
C
C
C      MULTIPLE INTERSECTION
370    WRITE (6,410) K,Q,R,T,RTHERM2
       DO 380 J=1,I
       WRITE (6,420) X(J),Y(J),X2(J)
380    CONTINUE
C
C      THE FOLLOWING IS A TEMPORARY APPROXIMATION FOR MULTIPLE INTERSECT.
       NPTS=0
       RETURN
C
390    FORMAT (75H OVERPRESSURE LOCUS LIES OUTSIDE OF AND DOES NOT CONTAI
      1N THE THERMAL CIRCLE)
400    FORMAT (70H  FROM SUBROUTINE EXTRACT,  OVERPRESSURE LOCUS CONTAINS
      1 THERMAL LOCUS )
410    FORMAT (1H-,33HMULTIPLE INTERSECTION WHERE  K = ,I3/1H ,4(E16.8,5X
      1)/)
420    FORMAT (1H ,E16.8,5X,E16.8,5X,F5.1)
       END
       FUNCTION BAKUP (R,T,TSPT)
       COMMON /DISTIME/ S(99,2),D(99,2),NUMDATA(2),CV(2),TI(2),ITYPE,NTYP
      1E,JTYPE,MTYPE,RADMAX
       INDEX=NUMDATA(ITYPE)
       IF (TSPT.LE.S(INDEX,ITYPE)) GO TO 10
C      WRITE(6,5) TSPT,NUMDATA(ITYPE)
       CALL TIMEGEN (TSPT)
10     INDEX=NUMDATA(ITYPE)
       DSPT=ALAG(TSPT,S(1,ITYPE),D(1,ITYPE),INDEX)
       RMAX=DSPT+R
       IF (RMAX.LE.D(INDEX,ITYPE)) GO TO 20
C      WRITE(6,15) RMAX,NUMDATA(ITYPE)
       CALL DATAGEN (RMAX)
20     INDEX=NUMDATA(ITYPE)
       TMAX=ALAG(RMAX,D(1,ITYPE),S(1,ITYPE),INDEX)
       T2=TMAX-T
       IF (T2.GE.TSPT) GO TO 30
C      WRITE(6,25) T2,TSPT
       BAKUP=0.0
       RETURN
30     INDEX=NUMDATA(ITYPE)
       BAKUP=ALAG(T2,S(1,ITYPE),D(1,ITYPE),INDEX)-DSPT
       IF (BAKUP.LT.0.0) BAKUP=0.0
       RETURN
```

```
C
      END
      SUBROUTINE DATAGEN (R)
      COMMON /DISTIME/ S(99,2),D(99,2),NUMDATA(2),CV(2),TI(2),ITYPE,NTYP
     1E,JTYPE,MTYPE,RADMAX
10    NUMDATA(ITYPE)=NUMDATA(ITYPE)+1
      INDEX=NUMDATA(ITYPE)
      S(INDEX,ITYPE)=S(INDEX-1,ITYPE)+TI(ITYPE)
      D(INDEX,ITYPE)=(CV(ITYPE)*TI(ITYPE))+D(INDEX-1,ITYPE)
      IF (R.GT.D(INDEX,ITYPE)) GO TO 10
      RETURN
      END
      SUBROUTINE TIMEGEN (TIME)
      COMMON /DISTIME/ S(99,2),D(99,2),NUMDATA(2),CV(2),TI(2),ITYPE,NTYP
     1E,JTYPE,MTYPE,RADMAX
10    NUMDATA(ITYPE)=NUMDATA(ITYPE)+1
      INDEX=NUMDATA(ITYPE)
      S(INDEX,ITYPE)=S(INDEX-1,ITYPE)+TI(ITYPE)
      D(INDEX,ITYPE)=(CV(ITYPE)*TI(ITYPE))+D(INDEX-1,ITYPE)
      IF (TIME.GT.S(INDEX,ITYPE)) GO TO 10
      RETURN
      END
      FUNCTION ALAG (XV,X,Y,NXY)
C
C     GENERAL PURPOSE FOUR-POINT LAGRANGIAN INTERPOLATION FUNCTION.
C     GIVES RESULT OF INTERPOLATION OF TABLE OF Y AS A FUNCTION OF X
C     AT ENTRY POINT XO.
C     NOTE - X ARRAY MUST BE IN ASCENDING ORDER.
C     FUNCTION BY HARRY M. MURPHY, JR., 30 NOVEMBER 1966.
C
      DIMENSION X(NXY), Y(NXY)
C
10    L=2
      M=NXY-1
      XO=XV
      IF (2-M) 20,80,80
C
20    I=(L+M)/2
      IF (L-I) 30,70,20
30    IF (X(I)-XO) 50,70,40
40    M=I
      GO TO 20
50    IF (X(I+1)-XO) 60,70,70
60    L=I
      GO TO 20
C
70    ALAG=Y(I-1)*(XO-X(I))*(XO-X(I+1))*(XO-X(I+2))/((X(I-1)-X(I))*(X(I-
     11)-X(I+1))*(X(I-1)-X(I+2)))+Y(I)*(XO-X(I-1))*(XO-X(I+1))*(XO-X(I+2
     2))/((X(I)-X(I-1))*(X(I)-X(I+1))*(X(I)-X(I+2)))+Y(I+1)*(XO-X(I-1))*
     3(XO-X(I))*(XO-X(I+2))/((X(I+1)-X(I-1))*(X(I+1)-X(I))*(X(I+1)-X(I+2
     4)))+Y(I+2)*(XO-X(I-1))*(XO-X(I))*(XO-X(I+1))/((X(I+2)-X(I-1))*(X(I
```

```
     5+2)-X(I))*(X(I+2)-X(I+1)))
      RETURN
C
80    WRITE (6,90) NXY,XO
      ALAG=0.0
      RETURN
C
C
90    FORMAT (26HOFUNCTION ALAG ERROR.  N =,I3,7H, ARG =,E12.4/1X)
      END
      SUBROUTINE SNAPTCM (IPROB2)
C
C     THE ORIGINAL VERSION, SNAPT, MENTIONED BELOW HAS BEEN MODIFIED BY
C     CRAIG E MILLER, CAPT, AFWL(SAB), KIRTLAND AFB, N. MEX., 247-711
C     EXT 2051 TO DETERMINE ONLY THE THERMAL RANGE SOLUTION WHICH IS ONE
C     SPECIFIC USE OF THE ORIGINAL MULTIPURPOSE PROGRAM.
C
C     PROGRAM SNAPT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,FILMPR,PUNCH)
C
C               SNAP-T WEAPON EFFECTS PROGRAM FOR THERMAL ENERGY
C
      COMMON /PASS/ SZ,AZ,DELA,AMAX,BETINO,BETA,W,RHO,VIS,PZ,HSL,HTE,HBL
     1,VEL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,DBURST,MBURST,ATM,CAL,TEFFI
      COMMON /TABLES/ XX(18),YY(5),ZZ(5,18),XXX(12),YYY(10),ZZZ(10,12),X
     1XXX(19),YYYY(20),ZZZZ(20,19),CX1(13),CX2(8),B(8,13)
      COMMON /INFO/ RS,WSA,CANGP,WSH,TD,ANG,FIR,FV,FW,DELU,DELL,THSTO,TAS
     1TD,FHBW,SLBR,SAVE,TEFF
      COMMON /INDCTR/ IND,K01
      REAL MBURST
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     KOUNT=1
C     1   CALL READ (KOUNT)
      SR=((W/CAL)**0.5)*5280.0
C
C     FOLLOWING RELATIVIZES A/C ALTITUDE TO SEA LEVEL BY ADDING GROUND
C     HEIGHT
      AZ=AZ+HTE
C
      DIFF=ABS(AZ-(HBL+HTE))
      IF (SR.LT.DIFF) SR=1.01*DIFF
      SZ=SQRT(SR**2-DIFF**2)
      DELA=AMAX=BETA=0.0
      VEL=ALPHL=TMPL=0.0
      WUAL=CPL=XLEL=0.0
      CRAFT=DBURST=0.0
      MBURST=0
      ATM=TEFFI=0.0
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      IF (TEFFI.EQ.0.) GO TO 10
      TEFF=TEFFI
      GO TO 20
```

```
10     TEFF=TEFU(W,HBL+HTE)                                            SNP 420
20     KO1=0                                                           SNP 430
       IND=0                                                           SNP 440
       WRITE (6,30)                                                    SNP 450
       WRITE (6,40) CRAFT,WUAL,VEL,AMAX,ALPHL,CPL,SZ,BETA,TMPL,XLEL,AZ,BESNP 460
      1TIND,DELA,CAL                                                   SNP 470
       WRITE (6,50) W,ATM,VIS,HBL,DAY,PZ,HTE,RHO,HSL,TEFF,DBURST,MBURST   SNP 480
       WRITE (6,60)                                                    SNP 490
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * SNP 500
C      CALL THERML                                                     SNP 510
       CALL THERML (IPROB2)                                            SNP 520
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * SNP 530
C      FOLLOWING RELATIVIZES A/C ALTITUDE TO GROUND                    SNP 540
       AZ=AZ-HTE                                                       SNP 550
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * SNP 560
       RETURN                                                          SNP 570
C      KOUNT=KOUNT+1                                                   SNP 580
C      GO TO 1                                                         SNP 590
C                                                                      SNP 600
C                                                                      SNP 610
C      4    FORMAT (1H1,135(1HX),//)                                   SNP 620
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * SNP 630
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * SNP 640
C      7    FORMAT (1X,135(1HX),///)                                   SNP 650
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * SNP 660
C                                                                      SNP 670
30     FORMAT (1H )                                                    SNP 680
40     FORMAT (6X,13HPANEL DATA   ,50X,19HRECEIVER PARAMETERS,//,6X,8HCRASNP 690
      1FT  =,E14.7,10X,8HWTL    =,E14.7,9X,8HFTSEC  =,E14.7,10X,8HMAXALT SNP 700
      2=,E14.7,/,6X,8HALPHAL =,E14.7,10X,8HBTUL    =,E14.7,9X,8HISTHR  =,ESNP 710
      314.7,10X,8HTILT    =,E14.7,/,6X,8HTMPL    =,E14.7,10X,8HXLEL    =,E14SNP 720
      4.7,9X,8HISTALT =,E14.7,10X,8HBETAID =,E14.7,/6X,8HDALT    =,E14.7,SNP 730
      510X,8HCAL     =,E14.7///)                                       SNP 740
50     FORMAT (6X,17HSOURCE PARAMETERS,46X,22HATMOSPHERIC PARAMETERS,//,6SNP 750
      1X,8HYIELD  =,E14.7,41X,8HATM     =,E14.7,10X,8HVISBLE =,E14.7,/,6X,SNP 760
      28HBURST  =,E14.7,41X,8HDAY     =,E14.7,10X,8HVAPOR  =,E14.7,/,6X,8HSNP 770
      3TARGET =,E14.7,41X,8HALBEDO =,E14.7,10X,8HHAZE   =,E14.7,/6X,8HTH SNP 780
      4EFF =,E14.7/6X,8HDBURST =,E14.7/6X,8HMBURST =,E14.7//)          SNP 790
60     FORMAT (1H )                                                    SNP 800
       END                                                             SNP 810-
       SUBROUTINE THERML(IPROB2)                                       THE  10
C      SUBROUTINE THERML                                               THE  20
       COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBLTHE  30
      1,VEL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,DBURST,MBURST,ATM,CAL,TEFFI   THE  40
       COMMON /NEEDED/ TP6,TM8,TM3,TP7,TM4,TM9,TM5,CON,PSL,TM6         THE  50
       COMMON /INFO/ RS,WSA,CANGP,WSH,TD,ANG,FIR,FV,FW,DELU,DELL,THSTD,TATHE  60
      1TD,FHBW,SLBR,SAVE,TEFF                                          THE  70
       COMMON /INDCTR/ IND,KO1                                         THE  80
       COMMON /CAN/ CON2,SZ1,MOM                                       THE  90
       COMMON /PROBLEM/ NEWPROB                                        THE 100
       LOGICAL NEWPROB                                                 THE 110
```

```
      REAL MBURST
      REAL X
      DATA TP10,TP9,TP8,TP7,TP6,TP5,TP4,TP3,TP2,TP1,T0,TM1,TM2,TM3,TM4,T
     1M5,TM6,TM7,TM8,TM9,TM10/1.0E+10,1.0E+9,1.0E+8,1.0E+7,1.0E+6,1.0E+5
     2,1.0E+4,1.0E+3,1.0E+2,1.0E+1,1.0,1.0E-1,1.0E-2,1.0E-3,1.0E-4,1.0E-
     35,1.0E-6,1.0E-7,1.0E-8,1.0E-9,1.0E-10/
      WSUBS(X)=2.3*PZ*(1.-10.**(-6.1*X*TM5))
      NEWPROB=.FALSE.
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * *
      KOUNT=0
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * *
      CON=-4.57*TM5
      SZ1=SZ
      DB=ABS(DBURST)
10    IF (AZ.LT.HTE) GO TO 520
      TST=HBL/W**(1.0/3.0)
      RS=238.1*W**(1.0/3.0)
      IF (TST-177.0) 20,30,30
20    THSTO=1.0
      TATO=0.0
      GO TO 60
30    IF (TST-369.0) 40,40,50
40    THSTO=6.21*TP6*((1.0/TST)**3.0-2.0*TM8)
      TATO=1.0-THSTO
      GO TO 60
50    THSTO=0.0
      TATO=1.0
60    IF (TST-183.0) 70,70,80
70    FHBW=3.41*TM3*TST+1.0
      GO TO 90
80    FHBW=1.63
90    IF (W-500.0) 100,100,110
100   DELL=1.0
      DELU=0.0
      GO TO 120
110   DELL=.9312-.2287*ALOG(W*TM3)*.43429
      DELU=1.0-DELL
120   WSH=WSUBS(HTE)
      FW=(3./2.)*(.32/TEFF)
130   IF (DBURST.GT.0.) AZ=HBL
      WSA=WSUBS(AZ)
      MOM=1
      IF (TST-244.0) 140,140,150
140   TD=3000.0+12.3*TST
      GO TO 160
150   TD=6000.0
160   FV=1.296*ALOG(TD)*.43429-4.426
      FIR=1.322-1.338*TM4*TD-1.170*TM9*TD**2
      QA=ABS(CAL)
170   HB=HBL
      SLBR=SZ
```

```
      CALL UPLOW (HB,SLBR,AL,BL,CBETL,CTHT,QNL)
      IF (NEWPROB) RETURN
      IF (K01.EQ.1) GO TO 440
      THT=ANG
      IF (DELU) 180,190,180
180   HB=HBL+RS
      CALL UPLOW (HB,SLBR,AU,BU,CBETU,CALP,QNU)
      IF (NEWPROB) RETURN
      IF (K01.EQ.1) GO TO 440
      ALP=ANG
190   CU=QNU*DELU
      CL=QNL*DELL
      IF (BETIND) 240,310,200
200   IF (DELU.EQ.0.0) GO TO 210
      BMU=ATAN(SQRT(1.-CBETU**2)/CBETU)
210   BML=ATAN(SQRT(1.-CBETL**2)/CBETL)
      IF ((ALP-BMU).LT.1.570796326) GO TO 220
      BU=0.
      IF ((THT-BML).LT.1.570796326) GO TO 220
      BL=0.
220   IF (RHO.EQ.0..AND.AZ.EQ.(HBL+HTE)) GO TO 230
      BETA=ATAN((AL*SQRT(1.0-CTHT**2)+BL*SQRT(1.0-CBETL**2)+CU/CL*(AU*SI
     1N(ALP)+BU*SQRT(1.0-CBETU**2)))/(AL*CTHT+BL*CBETL+CU/CL*(AU*CALP+BU
     2*CBETU)))
      IF (BETA.LT.0.) BETA=BETA+3.14159
      GO TO 310
230   BETA=1.57295
      GO TO 310
C     MODIFIED INCIDENT ANGLE PACKAGE
C
240   ANGLE=60.0/57.3
      SIDE=1.0
      IF (BETA-1.5708) 260,260,250
250   SIDE=2.0
260   IF (ALP-ANGLE) 280,280,270
270   ALP=(ALP+SIDE*ANGLE)/2.0
280   IF (THT-ANGLE) 300,300,290
290   THT=(THT+SIDE*ANGLE)/2.0
300   CONTINUE
C
310   IF (DELL.EQ.1.0) GO TO 320
      BMU=ATAN(SQRT(1.0-CBETU**2)/CBETU)
320   BML=ATAN(SQRT(1.0-CBETL**2)/CBETL)
      VERTU=COS(ALP-BETA)
      IF (VERTU) 330,330,340
330   VERTU=0.0
340   VERTL=COS(THT-BETA)
      IF (VERTL) 350,350,360
350   VERTL=0.0
360   QEUD=CU*AU*VERTU
      QEUR=CU*BU*(ABS(COS(BMU-BETA)))
```

```
      QELD=CL*AL*VERTL
      QELR=CL*BL*(ABS(COS(BML-BETA)))
      QEU=QEUD+QEUR
      QEL=QELD+QELR
      QE=QEL+QEU
      AZK=AZ/3281.
      HBK=HBL/3281.
      SZK=SZ/3281.
      SZN=SZ*.000164
      IF (CAL) 430,430,370
370   IF (ABS(QA-QE)-0.0001*QA) 380,480,490
380   IF (WUAL.LE.0.) GO TO 400
      IF (AZ.GT.(HBL-HTE+RS)) GO TO 390
      CALL TEMP (QE,TMP)
      GO TO 410
390   CALL TEMP (QE,TMP)
      GO TO 410
400   WRITE (6,530) AZ,HBL,AZK,HBK,SZ,SZK,BETA,SZN,CL,CU,QEL,QELD,QELR,Q
     1EU,QEUD,QEUR,QA,QE
      GO TO 420
410   WRITE (6,540) AZ,CL,SZ,CU,QEL,QELD,QELR,QEU,QEUD,QEUR,QA,QE,BETA,T
     1MP
420   MOM=1
      GO TO 440
430   WRITE (6,550) AZ,CL,SZ,CU,QEL,QELD,QELR,QEU,QEUD,QEUR,QE,BETA
      MOM=1
440   AZ=AZ+DELA
      IF (AZ.GT.AMAX.OR.AZ.EQ.0.) GO TO 450
      GO TO 130
450   HBL=HBL+DB
      IF (HBL.GT.HBURST.OR.HBL.EQ.0.) GO TO 470
      IF (TEFFI.GT.0.) GO TO 460
      TEFF=TEFU(W,HBL+HTE)
460   SZ=SZ1
      GO TO 10
470   RETURN
480   E=.3
      SSZ=(SZ*SZ*(QE/QA)**E+(AZ-HBL)*(AZ-HBL)*((QE/QA)**E-1.0))
      A=(QE/QA)**E
      IF (SSZ.LT.0.) GO TO 490
      SZ=SQRT(SSZ)
      SZN=SZ*.000164
      SZK=SZ/3281.
      MOM=MOM+1
      IF (MOM.GT.100) GO TO 510
      GO TO 170
490   IF (AZ.GE.(HBL+HTE)) GO TO 500
      WRITE (6,560) AZ
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     SZ=SZ1
C     GO TO 42
```

```
      IPROB2=1                                                          THE1650
      RETURN                                                            THE1660
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * THE1670
500   WRITE (6,570) AZ                                                  THE1680
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * THE1690
C     IF (DBURST.LT.0.) GO TO 43                                        THE1700
      IPROB2=1                                                          THE1710
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * THE1720
      RETURN                                                            THE1730
510   WRITE (6,580)                                                     THE1740
      RETURN                                                            THE1750
520   WRITE (6,590) AZ                                                  THE1760
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * *THE1770
      KOUNT=KOUNT+1                                                     THE1780
      IF (KOUNT.GE.10) STOP                                             THE1790
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * *THE1800
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * *THE1810
      AZ=AZ+HTE                                                         THE1820
      GO TO 10                                                          THE1830
C                                                                       THE1840
C                                                                       THE1850
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * THE1860
C     $Y AT CRITICAL PANEL   (CAL/CM**2)         QE =,E12.5//,1X,135(1H*),THE1870
C     $/)                                                               THE1880
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * THE1890
C                                                                       THE1900
530    FORMAT (6X,37HRECEIVER ALTITUDE            (FT) AZ =,E12.5,5X,27HBUTHE1910
     1RST ALTITUDE  (FT)  HBL =,E12.5/34X,9H(KM) AZ =,E12.5,21X,11H(KM) THE1920
     2 HBL =,E12.5/6X,37HCALCULATED HORIZONTAL RANGE (FT) SZ =,E12.5,5X,THE1930
     330HANGLE BETWEEN LOCAL HORIZONTAL/34X,9H(KM) SZ =,E12.5,5X,35HAND THE1940
     4CRITICAL PANEL (RADIANS) BETA =,E12.5/34X,9H(NM) SZ =,E12.5//,6X,5THE1950
     52HUNATTENUATED ENERGY IN LOWER PHASE (CAL/CM**2)  CL =,E12.5/6X,52THE1960
     6HUNATTENUATED ENERGY IN UPPER PHASE (CAL/CM**2)  CU =,E12.5,//,6X,THE1970
     765HATTENUATED ENERGY IN LOWER PHASE (CAL/CM**2)               QELTHE1980
     8 =,E12.5,4X,6HQELD =,E12.5,3X,6HQELR =,E12.5/,6X,65HATTENUATED ENETHE1990
     9RGY IN UPPER PHASE  (CAL/CM**2)               QEU =,E12.5,4X,6HQEUTHE2000
     $D =,E12.5,3X,6HQEUR =,E12.5//6X,64HTOTAL FREE FIELD ENERGY ATCRITITHE2010
     $CAL PANEL  (CAL/CM**2)         QA =,E12.5/6X,64HTOTAL ITERATED ENERGTHE2020
     $Y AT CRITICAL PANEL   (CAL/CM**2)         QE =,E12.5/)            THE2030
540    FORMAT (6X,37HAIRCRAFT ALTITUDE (FT)            AZ =,E12.5,5X,52HUNTHE2040
     1ATTENUATED ENERGY IN LOWER PHASE (CAL/CM**2)  CL =,E12.5,/,6X,37HCTHE2050
     2ALCULATED HORIZONTAL RANGE (FT) SZ =,E12.5,5X,52HUNATTENUATED ENERTHE2060
     3GY IN UPPER PHASE (CAL/CM**2)  CU =,E12.5,//,6X,65HATTENUATED ENERTHE2070
     4GY IN LOWER PHASE (CAL/CM**2)                QEL =,E12.5,4X,6HQELDTHE2080
     5 =,E12.5,3X,6HQELR =,E12.5,/,6X,65HATTENUATED ENERGY IN UPPER PHASTHE2090
     6E (CAL/CM**2)                 QEU =,E12.5,4X,6HQEUD =,E12.5,3X,6HQETHE2100
     7UR =,E12.5,//,6X,64HTOTAL FREE FIELD ENERGY AT CRITICAL PANEL (CALTHE2110
     8/CM**2)        QA =,E12.5,/,6X,64HTOTAL ITERATED ENERGY AT CRITICALTHE2120
     9 PANEL (CAL/CM**2)        QE =,E12.5,//,6X,66HANGLE BETWEEN LOCALTHE2130
     $ HORIZONTAL AND CRITICAL PANEL (RADIANS) BETA =,E12.5,/,6X,47H(USETHE2140
     $D FOR CALCULATION OF HORIZONTAL RANGE ONLY),//,66HCALCULATED TEMPETHE2150
```

```
     $RATURE OF CRITICAL PANEL (DEG-R)              TMP  =,E12.5,/,36H(ASS
     $UMING HORIZONTAL RECEIVER AT SZ),//,1X,135(1H/),///)
 550  FORMAT (6X,37HAIRCRAFT ALTITUDE (FT)           AZ =,E12.5,5X,52HUN
     1ATTENUATED ENERGY IN LOWER PHASE (CAL/CM**2)  CL =,E12.5,/,6X,37HH
     2ORIZONTAL RANGE (FT)            SZ =,E12.5,5X,52HUNATTENUATED ENER
     3GY IN UPPER PHASE (CAL/CM**2)  CU =,E12.5,//,6X,65HATTENUATED ENER
     4GY IN LOWER PHASE (CAL/CM**2)             QEL =,E12.5,4X,6HQELU
     5 =,E12.5,3X,6HQELR =,E12.5,/,6X,65HATTENUATED ENERGY IN UPPER PHAS
     6E (CAL/CM**2)             QEU =,E12.5,4X,6HQEUD =,E12.5,3X,6HQE
     7UR =,E12.5,//,6X,64HTOTAL FREE FIELD ENERGY AT CRITICAL PANEL (CAL
     8/CM**2)          QE =,E12.5,//,6X,66HANGLE BETWEEN LOCAL HORIZONTAL A
     9ND CRITICAL PANEL (RADIANS) BETA =,E12.5,//,1X,135(1H/),///)
 560  FORMAT (46H RECEIVER BELOW LOWER LIMIT OF ENVELOPE AT AZ=,F10.0,5H
     1  FT.)
 570  FORMAT (46H RECEIVER ABOVE UPPER LIMIT OF ENVELOPE AT AZ=,F10.0,5H
     1  FT.)
 580  FORMAT (21H SZ ITERATIONS GT 100)
 590  FORMAT (26H RECEIVER BELOW TERRAIN AT,F10.0,26H  FT,ALTITUDE INCR
     1MENTED.)
      END
      SUBROUTINE UPLOW (HB,SBR,A,B,CBET,CANG,QN)
      COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBL
     1,VEL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,JBURST,MBURST,ATM,CAL,TEFFI
      COMMON /INFO/ RS,WSA,CANGP,WSH,TD,ANG,FIR,FV,FW,DELU,DELL,THSTU,TA
     1TD,FHBW,SLBR,SAVE,TEFF
      COMMON /NEEDED/ TP6,TM8,TM3,TP7,TM4,TM9,TM5,CON,PSL,TM6
      COMMON /CAN/ CON2,SZ1,MOM
      COMMON /INDCTR/ IND,K01
      COMMON /PROBLEM/ NEWPROB
      LOGICAL NEWPROB
      REAL MBURST
      REAL X,Y,Z
      DSUBL(X)=SLBR**2+(AZ-X)**2
      COSAT(Y)=(AZ-Y)/SQRT(DSUBL(Y))
      WSUBS(Z)=2.3*PZ*(1.-10.**(-0.1*Z*TM5))
      TST=HB/W**(1.0/3.0)
      CANGP=COSAT(HTE)
      CANG=COSAT(HTE+HB)
      ANG=ACOS(CANG)
      IF (ANG) 10,20,20
 10   ANG=ANG+3.1416
 20   CONTINUE
      IF (TST-278.0) 30,40,40
 30   K=3
      GO TO 50
 40   K=2
 50   QN=QNFN(DSUBL(HTE+HB),CANG,K)
      TLUP=HTE+HB+.5*RS*CANG
      CALL PHASE (TH,TV,TPH,TPV,TLUP,CANG)
      IF (ABS(CANG).LT..001) GO TO 60
      WD=(WSA-WSUBS(TLUP))/CANG
```

```
      GO TO 70                                                          UPL 320
60    WD=1.615*TM4*PZ*(SLBR-RS)*(10.**(-6.1*TM5*TLUP)+10.**(-6.1*TM5*AZ)UPL 330
     1)                                                                 UPL 340
70    CALL TBLKUP (WD,TD,TW,1)                                          UPL 350
      IF (NEWPROB) RETURN                                               UPL 360
      IF (ABS(CANG).LT..001) GO TO 80                                   UPL 370
      IF (ABS(CANGP).LT..001) GO TO 100                                 UPL 380
      WDP=(WSA-WSH)/CANGP+WSUBS(HB)-WSH                                 UPL 390
      GO TO 110                                                         UPL 400
80    IF (ABS(CANGP).LT..001) GO TO 90                                  UPL 410
      WDP=(WSA-WSH)/CANGP+WSUBS(HB)-WSH                                 UPL 420
      GO TO 110                                                         UPL 430
90    WDP=1.615*TM4*PZ*((HB*(10.**(-6.1*TM5*HB)+10.**(-6.1*TM5*HTE)))+(SUPL 440
     1QRT(SLBR*SLBR+(AZ-HTE)*(AZ-HTE))*(10.**(-6.1*TM5*AZ)+10.**(-6.1*TMUPL 450
     25*HTE))))                                                         UPL 460
      GO TO 110                                                         UPL 470
100   WDP=1.615*TM4*PZ*SQRT(SLBR*SLBR+(AZ-HTE)*(AZ-HTE))*(10.**(-6.1*TM5UPL 480
     1*AZ)+10.**(-6.1*TM5*HTE))+WSUBS(HB)-WSH                           UPL 490
110   CALL TBLKUP (WDP,TD,TWP,1)                                        UPL 500
      IF (NEWPROB) RETURN                                               UPL 510
      IF (TST-278.0) 120,150,150                                        UPL 520
120   K=2                                                               UPL 530
      CON1=57.295*ANG                                                   UPL 540
      IF (CON1-90.0) 140,130,130                                        UPL 550
130   CON1=90.0                                                         UPL 560
140   CON2=SQRT(DSUBL(HTE+HB))/RS                                       UPL 570
      IF (CON2.LT.2.) GO TO 200                                         UPL 580
      GO TO 160                                                         UPL 590
150   K=3                                                               UPL 600
      CON1=(AZ-HTE)/HB                                                  UPL 610
      CON2=SLBR/HB                                                      UPL 620
160   CALL TBLKUP (CON2,CON1,GAM,K)                                     UPL 630
      IF (NEWPROB) RETURN                                               UPL 640
      IF (TST-273.0) 180,170,170                                        UPL 650
170   CBET=COSAT(HTE-HB)                                                UPL 660
      GO TO 190                                                         UPL 670
180   CBET=COSAT(HTE-3.0*RS)                                            UPL 680
190   A=FIR*TH*TW+FV*TV                                                 UPL 690
      B=(FIR*TPH*TWP+FV*TPV)*RHO*GAM/CBET                               UPL 700
      RETURN                                                            UPL 710
200   WRITE (6,210)                                                     UPL 720
      SZ=SZ1                                                            UPL 730
      K01=1                                                             UPL 740
      RETURN                                                            UPL 750
C                                                                       UPL 760
C                                                                       UPL 770
C                                                                       UPL 780
210   FORMAT (20H RANGE LESS THAN 2RS)                                  UPL 790
      END                                                               UPL 800-
      SUBROUTINE PHASE (TH,TV,TPH,TPV,TLUP,CANG)                        PHA  10
      COMMON /PASS/ SZ,AZ,DELA,AMAX,BETINO,BETA,W,RHO,VIS,PZ,HSL,HTE,HBLPHA  20
```

```
     1,VEL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,DBURST,MBURST,ATM,CAL,TEFFI
      COMMON /INFO/ RS,WSA,CANGP,WSH,TO,ANG,FIR,FV,FW,DELU,DELL,THSTO,TA
     1TO,FHBW,SLBR,SAVE,TEFF
      COMMON /NEEDED/ TP6,TM8,TM3,TP7,TM4,TM9,TM5,CON,PSL,TM6
      REAL MBURST
      REAL X,Y,Z,ZZ
      EXPON(X,Y,Z,ZZ)=EXP(X*(EXP(CON*Y)-EXP(CON*Z))/ZZ)
      EXPON1(X,Y,Z,ZZ)=EXP(X*(EXP(CON*Y)+EXP(CON*Z))/ZZ)
      IF (ABS(CANG).LE..001) GO TO 80
      IF (TLUP.LT.HSL) GO TO 40
      IF (AZ.LT.TLUP) GO TO 10
      THAZE=1.
      TVIS=EXPON(-.0875,TLUP,AZ,CANG)
      TPHAZE=EXPON(-16.4/VIS,HTE,HSL,CANGP)
      TPVIS=EXPON(-.0875,HSL,AZ,CANGP)
      GO TO 120
10    IF (AZ.LT.HSL) GO TO 20
      THAZE=1.
      TVIS=EXPON(.0875,AZ,TLUP,CANG)
      TPHAZE=EXPON(-16.4/VIS,HTE,HSL,CANGP)
      TPVIS=EXPON(-.0875,HSL,AZ,CANGP)
      GO TO 120
20    IF (ABS(CANGP).LE..001) GO TO 30
      THAZE=EXPON(16.4/VIS,AZ,HSL,CANG)
      TVIS=EXPON(.0875,HSL,TLUP,CANG)
      TPHAZE=EXPON(-16.4/VIS,HTE,AZ,CANGP)
      TPVIS=1.
      GO TO 120
30    THAZE=EXPON(16.4/VIS,AZ,HSL,CANG)
      TVIS=EXPON(.0875,HSL,TLUP,CANG)
      TPHAZE=EXPON1(-3.75*TM4*SLBR/VIS,HTE,AZ,1.)
      TPVIS=1.
      GO TO 120
40    IF (AZ.LT.TLUP) GO TO 60
      IF (AZ.GT.HSL) GO TO 50
      THAZE=EXPON(-16.4/VIS,TLUP,AZ,CANG)
      TVIS=1.
      TPHAZE=EXPON(-16.4/VIS,HTE,AZ,CANGP)
      TPVIS=1.
      GO TO 120
50    THAZE=EXPON(-16.4/VIS,TLUP,HSL,CANG)
      TVIS=EXPON(-.0875,HSL,AZ,CANG)
      TPHAZE=EXPON(-16.4/VIS,HTE,HSL,CANGP)
      TPVIS=EXPON(-.0875,HSL,AZ,CANGP)
      GO TO 120
60    IF (ABS(CANGP).LE..001) GO TO 70
      THAZE=EXPON(16.4/VIS,AZ,TLUP,CANG)
      TVIS=1.
      TPHAZE=EXPON(-16.4/VIS,HTE,AZ,CANGP)
      TPVIS=1.
      GO TO 120
```

```
70    THAZE=EXPON(16.4/VIS,AZ,TLUP,CANG)
      TVIS=1.
      TPHAZE=EXPON1(-3.75*TM4*SLBR/VIS,AZ,HTE,1.)
      TPVIS=1.
      GO TO 120
80    IF (ABS(CANGP).LE..001) GO TO 100
      IF ((AZ+TLUP)/2..LT.HSL) GO TO 90
      THAZE=1.0
      TVIS=EXPON1(-2.*TM6*(SLBR-RS),AZ,TLUP,1.)
      TPHAZE=EXPON(-16.4/VIS,HTE,HSL,CANGP)
      TPVIS=EXPON(-.0875,HSL,AZ,CANGP)
      GO TO 120
90    THAZE=EXPON1(-3.75*TM4*(SLBR-RS)/VIS,TLUP,AZ,1.)
      TVIS=1.0
      TPHAZE=EXPON(-16.4/VIS,HTE,AZ,CANGP)
      TPVIS=1.0
      GO TO 120
100   IF ((AZ+TLUP)/2..LT.HSL) GO TO 110
      THAZE=1.
      TVIS=EXPON1(-2.*TM6*(SLBR-RS),AZ,TLUP,1.)
      TPHAZE=1.
      TPVIS=EXPON1(-2.*TM6*SLBR,AZ,TLUP,1.)
      GO TO 120
110   THAZE=EXPON1(-3.75*TM4*(SLBR-RS)/VIS,TLUP,AZ,1.)
      TVIS=1.
      TPHAZE=EXPON1(-3.75*TM4*SLBR/VIS,HTE,AZ,1.)
      TPVIS=1.
120   TH=THAZE*.3+.7
      TV=.3*THAZE*TVIS+.7
      TPH=.3*TPHAZE+.7
      TPV=.3*TPHAZE*TPVIS+.7
      RETURN
      END
      FUNCTION QNFN (X,Y,K)
      COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBL
     1,VEL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,DBURST,MBURST,ATM,CAL,TEFFI
      COMMON /NEEDED/ TP6,TM6,TM3,TP7,TM4,TM9,TM5,CON,PSL,TM6
      COMMON /INFO/ RS,WSA,CANGP,WSH,TD,ANG,FIR,FV,FW,DELU,DELL,THST0,TA
     1T0,FHBW,SLBR,SAVE,TEFF
      REAL MBURST
      TEMP=8.569992*TEFF*W*TP7/X
      IF (K-2) 20,20,10
10    CUN=2.0/3.0*FW
      GO TO 30
20    CUN=1.0
30    QNFN=TEMP*(TAT0+CUN*THST0*FHBW*COS(2.0/3.0*ANG))
      RETURN
      END
      SUBROUTINE TBLKUP (X,Y,Z,K)
      COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBL
     1,VEL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,DBURST,MBURST,ATM,CAL,TEFFI
```

```
      COMMON /INFO/ RS,WSA,CANGP,WSH,TO,ANG,FIR,FV,FW,DELU,DELL,THSTO,TA
     1TO,FHBW,SLBR,SAVE,TEFF
      COMMON /TABLES/ XX(18),YY(5),ZZ(5,18),XXX(12),YYY(10),ZZZ(10,12),X
     1XXX(19),YYYY(20),ZZZZ(20,19),CX1(13),CX2(8),B(8,13)
      COMMON /NEEDED/ TP6,TM8,TN3,TP7,TM4,TM9,TM5,CON,PSL,TM6
      DIMENSION X1(2)
      REAL MBURST
   10 GO TO (20,30,40,50,60), K
   20 X1(1)=X
      X1(2)=Y
      CALL DINT (XX,YY,ZZ,5,18,X1(1),X1(2),Z)
      GO TO 70
   30 X1(1)=X
      X1(2)=Y
      CALL DINT (XXX,YYY,ZZZ,10,12,X1(1),X1(2),Z)
      GO TO 70
   40 X1(1)=X
      X1(2)=Y
      CALL DINT (XXXX,YYYY,ZZZZ,20,19,X1(1),X1(2),Z)
      GO TO 70
   50 X1(1)=X
      X1(2)=Y
      CALL DINT (CX1,CX2,B,8,13,X1(1),X1(2),Z)
      GO TO 70
   60 X1(1)=X
      X1(2)=Y
      CALL DINT (XX1,XY1,XZ1,19,2,X1(1),X1(2),Z)
   70 RETURN
      END
      FUNCTION TEFU (YIELD,HOB)
C     THIS FUNCTION PROVIDES THE UNATTENUATED THERMAL EFFICIENCY AT THE
C     SPECIFIC YIELD AND BURST ALTITUDE
      H=HOB*3.048E-4
      TEFU=EXP((-3.5797123034E-1-8.8040573390E-3*H+7.1368010068E-4*H*H-1
     1.2548009745E-5*H*H*H+6.4232050535E-8*H*H*H*H)*2.302585093)
      RETURN
      END
      SUBROUTINE ATMOSP (AZ,TA,PA,CA,SSG,ROSL,ATM)
   10 ROSL=.002378
      IF (AZ-36089.0) 20,20,30
   20 TA=518.688-.00356616*AZ
      PA=29.9213*(TA/518.688)**5.2561393
      CA=49.040895*SQRT(TA)
      SSG=(1.0-.00000689*AZ)**(-2.128)
      GO TO 40
   30 TA=389.988
      PA=6.8322*EXP(-4.8063618*(10.0**(-5))*(AZ-36089.0))
      CA=968.405
      SSG=1.8146*EXP(2.4032*(AZ-36089.0)*10.0**(-5.0))
   40 CONTINUE
      RETURN
```

```
      END
      SUBROUTINE TEMP (Q,TMP)
C     CALCULATES MAXIMUM TEMPERATURE FOR A HORIZONTAL PANEL SUBJECTED TO
C     A TOTAL FLUX OF Q CAL/CM**2 IMPINGING WITH ANGLE BETA
      COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBL
     1,VEL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,DBURST,MBURST,ATM,CAL,TEFFI
      COMMON /NEEDED/ TP6,TM8,TM3,TP7,TM4,TM9,TM5,CON,PSL,TM6
      REAL MBURST
      IF (BETA.GT.1.57) GO TO 40
      CALL ATMOSP (AZ,TA,PA,CA,SSGA,ROSL,ATM)
      SMLM=ABS(VEL/CA)
      TBL=TA*(1.+.18*SMLM**2)
      TF=(TBL+TA)/2.0
      HA=5.46667*TM3*(SMLM*PA)**.8*TA**.4/(XLEL**.2*TF**.545)
      TAU=.248*HA*SQRT(W)/(CPL*WUAL)
      IF (TAU-.2) 10,10,20
10    TPMAX=1.0134-.7147*TAU
      GO TO 30
20    TPMAX=.627-.362*ALOG(TAU)*.43429
30    QE=Q*COS(BETA)
      QAA=QE*ALPHL
      TMP=TBL+QAA*TPMAX/(.27115*WUAL*CPL)
      RETURN
40    WRITE (6,50)
      RETURN
C
C
C
50    FORMAT (22H RECEIVER BELOW SOURCE)
      END
      SUBROUTINE DINT (XX,YY,ZZ,M,N,X,Y,Z)
C     LINEAR 2 DIMENSIONAL INTERPOLATION
C     EXAMPLE-GIVEN THE FOLLOWING TABLE,AND X AND Y VALUES,WE CAN
C     1INTERPOLATE FOR APPROPIATE Z VALUE
C     XX AND YY MUST BE IN INCREASING ORDER
C        X1  X2  X3
C     1Y1 Z11 Z12 Z13
C     2Y2 Z21 Z22 Z23
C     3Y3 Z31 Z32 Z33
      COMMON /PROBLEM/ NEWPROB
      DIMENSION XX(N), YY(M), ZZ(M,N)
      LOGICAL NEWPROB
      IF (((X.LT.XX(1)).OR.(X.GT.XX(N))).OR.((Y.LT.YY(1)).OR.(Y.GT.YY(M)
     1))) GO TO 70
      MM=M-1
      NN=N-1
      DO 20 I=1,MM
      IF ((YY(I).LE.Y).AND.(YY(I+1).GE.Y)) GO TO 10
      GO TO 20
10    C1=(Y-YY(I))/(YY(I+1)-YY(I))
      I1=I
```

```
      I2=I+1
      GO TO 30
   20 CONTINUE
   30 DO 50 J=1,NN
      IF ((XX(J).LE.X).AND.(XX(J+1).GE.X)) GO TO 40
      GO TO 50
   40 C2=(X-XX(J))/(XX(J+1)-XX(J))
      J1=J
      J2=J+1
      GO TO 60
   50 CONTINUE
   60 CONTINUE
      Z1=ZZ(I1,J1)+C1*(ZZ(I2,J1)-ZZ(I1,J1))
      Z2=ZZ(I1,J2)+C1*(ZZ(I2,J2)-ZZ(I1,J2))
      Z=Z1+C2*(Z2-Z1)
      RETURN
   70 WRITE (6,80) X,Y
      NEWPROB=.TRUE.
      RETURN
C
C
C
   80 FORMAT (1H1,26HPARAMETER OUTSIDE RANGE,X=,E15.8,2X,2HY=,E15.6)
      END
      BLOCK DATA
      COMMON /TABLES/ XX(18),YY(5),ZZ(5,18),XXX(12),YYY(10),ZZZ(10,12),X
     1XXX(19),YYYY(20),ZZZZ(20,19),CX1(13),CX2(8),B(8,13)
      DATA XX/0.,2.00000E+00,4.00000E+00,6.00000E+00,8.00000E+00,1.00000
     1E+01,1.50000E+01,2.00000E+01,2.50000E+01,3.00000E+01,4.00000E+01,5
     2.00000E+01,7.00000E+01,9.00000E+01,1.10000E+02,1.50000E+02,3.00000
     3E+02,1.00000E+03/
      DATA YY/2.00000E+03,3.00000E+03,4.00000E+03,5.00000E+03 6.00000E+0
     13/
      DATA ZZ/8.45000E-01,9.05000E-01,9.43000E-01,9.70000E-01,9.75000E-0
     11,7.40000E-01,8.40000E-01,8.80000E-01,8.82000E-01,9.10000E-01,6.83
     2000E-01,7.84000E-01,8.17000E-01,8.43000E-01,8.62000E-01,6.53000E-0
     31,7.59000E-01,8.03000E-01,8.20000E-01,8.39000E-01,6.30000E-01,7.38
     4000E-01,7.81000E-01,7.98000E-01,8.13000E-01,6.10000E-01,7.18000E-0
     51,7.60000E-01,7.80000E-01,8.00000E-01,5.75000E-01,6.80000E-01,7.22
     6000E-01,7.45000E-01,7.60000E-01,5.51000E-01,6.50000E-01,6.98000E-0
     71,7.17000E-01,7.33000E-01,5.35000E-01,6.28000E-01,6.74000E-01,6.98
     8000E-01,7.15000E-01,5.18000E-01,6.11000E-01,6.48000E-01,6.76000E-0
     91,6.98000E-01,4.94000E-01,5.81000E-01,6.23000E-01,6.48000E-01,6.62
     $000E-01,4.72000E-01,5.59000E-01,6.01000E-01,6.21000E-01,6.38000E-0
     $1,4.28000E-01,5.19000E-01,5.58000E-01,5.79000E-01,5.92000E-01,4.06
     $000E-01,4.82000E-01,5.20000E-01,5.41000E-01,5.55000E-01,3.80000E-0
     $1,4.52000E-01,4.90000E-01,5.10000E-01,5.21000E-01,3.40000E-01,4.02
     $000E-01,4.40000E-01,4.55000E-01,4.65000E-01,2.40000E-01,2.60000E-0
     $1,2.90000E-01,2.50000E-01,2.70000E-01,1.00000E-01,1.00000E-01,1.00
     $000E-01,1.00000E-01,1.00000E-01/
      DATA XXX/2.00000E+00,3.00000E+00,5.00000E+00,7.50000E+00,1.00000E+
```

```
101,1.50000E+01,2.00000E+01,2.50000E+01,3.00000E+01,4.00000E+01,5.0
20000E+01,1.00000E+02/
 DATA YYY/0.,1.00000E+01,2.00000E+01,3.00000E+01,4.00000E+01,5.0000
10E+01,6.00000E+01,7.00000E+01,8.00000E+01,9.00000E+01/
 DATA (ZZZ(I),I=1,60)/9.58800E-02,1.03600E-01,1.13180E-01,1.29070E-
101,1.43710E-01,1.56880E-01,1.63820E-01,1.59600E-01,1.38400E-01,1.1
25000E-01,1.98000E-01,1.98570E-01,1.98370E-01,1.95520E-01,1.90490E-
301,1.80120E-01,1.60830E-01,1.31150E-01,9.58700E-02,7.30000E-02,3.0
47130E-01,3.00440E-01,2.81630E-01,2.55980E-01,2.24740E-01,1.67740E-
501,1.45610E-01,1.01420E-01,6.25000E-02,4.30000E-02,3.66830E-01,3.5
67430E-01,3.28430E-01,2.88330E-01,2.40640E-01,1.88200E-01,1.34600E-
701,8.48700E-02,4.59100E-02,2.80000E-02,4.00930E-01,3.87260E-01,3.5
83020E-01,3.05000E-01,2.48740E-01,1.87770E-01,1.28520E-01,7.62200E-
902,3.75600E-02,2.10000E-02,4.33620E-01,4.17940E-01,3.78340E-01,3.2
$1970E-01,2.55870E-01,1.87000E-01,1.21960E-01,6.73000E-02,2.91800E-
$02,1.40000E-02/
 DATA (ZZZ(I),I=61,120)/4.50130E-01,4.33600E-01,3.91250E-01,3.30760
1E-01,2.59620E-01,1.86370E-01,1.18480E-01,6.27400E-02,2.49800E-02,1
2.10000E-02,4.60080E-01,4.43090E-01,3.99060E-01,3.35770E-01,2.61970
3E-01,1.85990E-01,1.16360E-01,5.99800E-02,2.24200E-02,8.00000E-03,4
4.66720E-01,4.49450E-01,4.04370E-01,3.39300E-01,2.63340E-01,1.85720
5E-01,1.14940E-01,5.81200E-02,2.07300E-02,7.00000E-03,4.75100E-01,4
6.57510E-01,4.10960E-01,3.43360E-01,2.65210E-01,1.85340E-01,1.13120
7E-01,5.57800E-02,1.86100E-02,5.00000E-03,4.80090E-01,4.62320E-01,4
8.14930E-01,3.46290E-01,2.66320E-01,1.85120E-01,1.12030E-01,5.4360
9E-02,1.73200E-02,4.00000E-03,4.90010E-01,4.71930E-01,4.22860E-01,3
$.51510E-01,2.68490E-01,1.84560E-01,1.09800E-01,5.15400E-02,1.47600
$E-02,2,00000E-03/
 DATA XXXX/0.,1.00000E+00,2.00000E+00,3.00000E+00,4.00000E+00,5.000
100E+00,6.00000E+00,8.00000E+00,1.00000E+01,1.20000E+01,1.40000E+01
2,1.60000E+01,1.80000E+01,2.00000E+01,2.20000E+01,2.60000E+01,3.000
300E+01,3.50000E+01,4.00000E+01/
 DATA YYYY/0.,2.50000E-01,5.00000E-01,7.50000E-01,1.00000E+00,1.500
100E+00,2.00000E+00,3.00000E+00,4.00000E+00,6.00000E+00,8.00000E+00
2,1.00000E+01,1.20000E+01,1.40000E+01,1.80000E+01,2.20000E+01,2.600
300E+01,3.00000E+01,3.50000E+01,4.00000E+01/
 DATA (ZZZZ(I),I=1,95)/1.00000E+00,4.94300E-01,1.64300E-01,4.29000E
1-02,1.00000E-03,6.71000E-02,1.99000E-01,4.53400E-01,6.67000E-01,9.
272000E-01,1.16700E+00,1.31000E+00,1.41000E+00,1.48500E+00,1.59400E
3+00,1.66900E+00,1.72200E+00,1.76400E+00,1.81000E+00,1.83500E+00,7.
407000E-01,5.37500E-01,4.19600E-01,3.03400E-01,2.69300E-01,2.60900E
5-01,3.11000E-01,5.03400E-01,6.86100E-01,9.79000E-01,1.16000E+00,1.
630000E+00,1.40000E+00,1.48000E+00,1.59200E+00,1.66800E+00,1.72000E
7+00,1.76200E+00,1.80600E+00,1.83200E+00,4.47000E-01,4.31400E-01,4.
836600E-01,4.30200E-01,4.37200E-01,4.48000E-01,4.72400E-01,4.57100E
9-01,7.20600E-01,9.63000E-01,1.14900E+00,1.28000E+00,1.38200E+00,1.
$46000E+00,1.58200E+00,1.66000E+00,1.71200E+00,1.75600E+00,1.79900E
$+00,1.82300E+00,3.16000E-01,3.22600E-01,3.43500E-01,3.72400E-01,4.
$00400E-01,4.56300E-01,5.08500E-01,6.17100E-01,7.31000E-01,9.42500E
$-01,1.12300E+00,1.25800E+00,1.35000E+00,1.44000E+00,1.56600E+00,1.
$646.0E+00,1.70200E+00,1.74900E+00,1.79000E+00,1.81900E+00,2.43000E
```

```
$-01,2.49000E-01,2.70000E-01,2.96000E-01,3.29000E-01,3.95700E-01,4.BD1 790
$62500E-01,5.84800E-01,7.03700E-01,9.08200E-01,1.08600E+00,1.19100EBD1 800
$+00,1.32600E+00,1.41000E+00,1.54200E+00/                          BD1 810
 DATA (ZZZZ(I),I=96,190)/1.64000E+00,1.68600E+00,1.73700E+00,1.7800BD1 820
10E+00,1.81000E+00,1.96200E-01,2.00000E-01,2.17700E-01,2.36000E-01,BD1 830
22.67200E-01,3.26500E-01,3.95700E-01,5.26300E-01,6.35000E-01,8.6000BD1 840
30E-01,1.03800E+00,1.18000E+00,1.29200E+00,1.38100E+00,1.52800E+00,BD1 850
41.61000E+00,1.67200E+00,1.72200E+00,1.77000E+00,1.79500E+00,1.6440BD1 860
50E-01,1.71000E-01,1.80600E-01,2.00000E-01,2.20100E-01,2.73700E-01,BD1 870
63.34400E-01,4.60500E-01,5.72200E-01,8.00000E-01,9.81100E-01,1.1270BD1 880
70E+00,1.24100E+00,1.34100E+00,1.48600E+00,1.57900E+00,1.64900E+00,BD1 890
81.70600E+00,1.75300E+00,1.78400E+00,1.24000E-01,1.29000E-01,1.3400BD1 900
90E-01,1.46000E-01,1.59900E-01,1.96500E-01,2.42900E-01,3.44600E-01,BD1 910
$4.55000E-01,6.67400E-01,8.52700E-01,1.00600E+00,1.13300E+00,1.2450BD1 920
$0E+00,1.49900E+00,1.52000E+00,1.60000E+00,1.66300E+00,1.72100E+00,BD1 930
$1.76000E+00,9.95000E-02,1.00000E-01,1.06500E-01,1.15000E-01,1.2370BD1 940
$0E-01,1.48500E-01,1.82000E-01,2.62000E-01,3.53600E-01,5.45700E-01,BD1 950
$7.26900E-01,8.84100E-01,1.02200E+00,1.14000E+00,1.32100E+00,1.4510BD1 960
$0E+00,1.54700E+00,1.62100E+00,1.69000E+00,1.73500E+00,8.30000E-02,BD1 970
$8.50000E-02,8.90000E-02,9.10000E-02,1.00000E-01,1.17000E-01,1.4300BD1 980
$0E-01,2.05000E-01,2.78000E-01,4.41000E-01/                        BD1 990
 DATA (ZZZZ(I),I=191,285)/6.10000E-01,7.68000E-01,9.08000E-01,1.028BD11000
100E+00,1.22800E+00,1.37000E+00,1.47800E+00,1.56200E+00,1.64900E+00BD11010
2,1.70100E+00,7.12000E-02,7.20000E-02,7.56000E-02,8.00000E-02,8.460BD11020
300E-02,9.82000E-02,1.16600E-01,1.64700E-01,2.24900E-01,3.66400E-01BD11030
4,5.17300E-01,6.65400E-01,8.00000E-01,9.24400E-01,1.13100E+00,1.286BD11040
500E+00,1.40100E+00,1.50300E+00,1.60000E+00,1.66300E+00,6.24000E-02BD11050
6,6.30000E-02,6.50000E-02,6.90000E-02,7.20000E-02,8.40000E-02,9.800BD11060
700E-02,1.38000E-01,1.85000E-01,3.07000E-01,4.39000E-01,5.71000E-01BD11070
8,7.06000E-01,8.30000E-01,1.04000E+00,1.20300E+00,1.32900E+00,1.432BD11080
900E+00,1.54300E+00,1.61700E+00,5.55000E-02,5.60000E-02,5.67000E-02BD11090
$,6.00000E-02,6.43000E-02,7.23000E-02,8.38000E-02,1.15300E-01,1.550BD11100
$00E-01,2.60000E-01,3.74500E-01,4.98000E-01,6.20000E-01,7.38300E-01BD11110
$,9.46700E-01,1.12000E+00,1.25200E+00,1.37000E+00,1.48100E+00,1.567BD11120
$00E+00,4.98000E-02,5.00000E-02,5.05000E-02,5.20000E-02,5.65000E-02BD11130
$,6.90000E-02,7.25000E-02,9.30000E-02,1.31000E-01,2.19000E-01,3.210BD11140
$00E-01,4.32000E-01,5.50000E-01,6.56000E-01,8.61000E-01,1.04000E+00BD11150
$,1.18100E+00,1.30000E+00,1.43700E+00,1.51600E+00,4.54000E-02,4.540BD11160
$00E-02,4.54000E-02,4.90000E-02,5.08000E-02/                      BD11170
 DATA (ZZZZ(I),I=286,380)/5.65000E-02,6.46000E-02,8.62000E-02,1.148BD11180
100E-01,1.89500E-01,2.79900E-01,3.80200E-01,4.88000E-01,5.89800E-01BD11190
2,7.87400E-01,9.61400E-01,1.11800E+00,1.23400E+00,1.35900E+00,1.463BD11200
300E+00,3.64000E-02,3.90000E-02,4.00000E-02,4.10000E-02,4.30000E-02BD11210
4,4.60000E-02,5.60000E-02,6.70000E-02,8.90000E-02,1.42000E-01,2.180BD11220
500E-01,2.99000E-01,3.88000E-01,4.82000E-01,6.52000E-01,8.26000E-01BD11230
6,9.83000E-01,1.10900E+00,1.23800E+00,1.35700E+00,3.33000E-02,3.300BD11240
700E-02,3.30000E-02,3.40000E-02,3.57000E-02,3.93000E-02,4.37000E-02BD11250
8,5.56000E-02,7.17000E-02,1.15200E-01,1.71400E-01,2.37200E-01,3.080BD11260
900E-01,3.88100E-01,5.46900E-01,7.04200E-01,8.52000E-01,9.81500E-01BD11270
$,1.12300E+00,1.24100E+00,2.86000E-02,2.80000E-02,2.80000E-02,2.800BD11280
$00E-02,2.90000E-02,3.30000E-02,3.60000E-02,4.10000E-02,5.80000E-02BD11290
```

```
$,9.00000E-02,1.32000E-01,1.90000E-01,2.52000E-01,3.18000E-01,4.570BD11300
$00E-01,6.03000E-01,7.43000E-01,8.48000E-01,9.96000E-01,1.11800E+00BD11310
$,2.50000E-02,2.50000E-02,2.50000E-02,2.50000E-02,2.61000E-02,2.820BD11320
$00E-02,3.08000E-02,3.76000E-02,4.70000E-02,7.24000E-02,1.06400E-01BD11330
$,1.47700E-01,1.92000E-01,2.47900E-01,3.63000E-01,4.85500E-01,6.090BD11340
$00E-01,7.29300E-01,8.70000E-01,1.00100E+00/                       BD11350
 DATA CX1/0.,5.00000E+00,6.00000E+00,7.00000E+00,8.00000E+00,9.0000BD11360
10E+00,1.00000E+01,1.50000E+01,2.00000E+01,2.50000E+01,3.50000E+01,BD11370
24.50000E+01,9.00000E+01/                                           BD11380
 DATA CX2/0.,5.00000E+03,1.00000E+04,1.50000E+04,2.00000E+04,2.5000BD11390
10E+04,3.00000E+04,6.00000E+04/                                     BD11400
 DATA (B(I),I=1,50)/0.,0.,0.,0.,0.,0.,0.,0.,5.00000E+00,1.18000E+01BD11410
1,1.72000E+01,2.28000E+01,2.80000E+01,3.40000E+01,3.74000E+01,5.000BD11420
200E+01,6.00000E+00,1.18000E+01,1.63000E+01,2.12000E+01,2.57000E+01BD11430
3,3.13000E+01,3.48000E+01,4.60000E+01,7.00000E+00,1.20000E+01,1.590BD11440
400E+01,2.06000E+01,2.41000E+01,2.92000E+01,3.28000E+01,4.45000E+01BD11450
5,8.50000E+00,1.24000E+01,1.59000E+01,1.97000E+01,2.31000E+01,2.760BD11460
600E+01,3.12000E+01,4.29000E+01,9.30000E+00,1.30000E+01,1.61000E+01BD11470
7,1.95000E+01,2.26000E+01,2.63000E+01,2.99000E+01,4.19000E+01,1.000BD11480
800E+01,1.36000E+01/                                                BD11490
 DATA (B(I),I=51,104)/1.65000E+01,1.95000E+01,2.23000E+01,2.55000E+BD11500
101,2.90000E+01,4.10000E+01,1.50000E+01,1.74000E+01,1.92000E+01,2.1BD11510
25000E+01,2.35000E+01,2.58000E+01,2.80000E+01,4.05000E+01,2.00000E+BD11520
301,2.17000E+01,2.31000E+01,2.48000E+01,2.61000E+01,2.81000E+01,2.9BD11530
48000E+01,3.90000E+01,2.50000E+01,2.65000E+01,2.73000E+01,2.86000E+BD11540
501,2.96000E+01,3.12000E+01,3.25000E+01,3.96000E+01,3.50000E+01,3.6BD11550
60000E+01,3.67000E+01,3.74000E+01,3.83000E+01,3.90000E+01,3.98000E+BD11560
701,4.46000E+01,4.50000E+01,4.59000E+01,4.65000E+01,4.69000E+01,4.7BD11570
85000E+01,4.82000E+01,4.96000E+01,5.22000E+01,9.00000E+01,9.00000E+BD11580
901,9.00000E+01,9.00000E+01,9.00000E+01,9.00000E+01,9.00000E+01,9.0BD11590
$0000E+01/                                                          BD11600
 END                                                                BD11610
 SUBROUTINE SABERCM (SRCM,HZCM,HGCM,HBCM,WCM,DELPCM,HORF,TSA,NCASE,SAB  10
1IPROB1)                                                           SAB  20
C                                                                  SAB  30
C    THE ORIGINAL VERSION, SABER, DESCRIBED BELOW HAS BEEN MODIFIED BY SAB  40
C    CRAIG E MILLER, CAPT, AFWL(SAB), KIRTLAND AFB, N. MEX., 2471711   SAB  50
C    EXT 2051 TO DETERMINE ONLY THE OVERPRESSURE RANGE SOLUTION AND    SAB  60
C    TIME OF SHOCK ARRIVAL WHICH IS ONE SPECIFIC USE OF THE ORIGINAL   SAB  70
C    MULTIPURPOSE PROGRAM.  OCTOBER 1972                               SAB  80
C                                                                      SAB  90
C    PROGRAM SABER(INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT)        SAB 100
C    ******************************************************************SAB 110
C    THIS PROGRAM CALCULATES THE FREE-FIELD BLAST PARAMETERS ASSOCIATEDSAB 120
C    WITH THE DETONATION OF A NUCLEAR WEAPON.  THE MODEL USES          SAB 130
C    SIMILITUDE TECHNIQUES BASED ON THE IBM M-PROBLEM 1-KT SEA LEVEL   SAB 140
C    CALCULATION.  THE ORIGINAL VERSION OF THE PROGRAM WAS WRITTEN BY  SAB 150
C    ARTURO V. SERRANO OF TECHNOLOGY INC.  THE PRESENT VERSION IS THE  SAB 160
C    RESULT OF AN EFFORT TO EXPAND THE CAPABILITY OF THE CODE AND      SAB 170
C    WAS WRITTEN BY CAPT W. R. DASSOW AND MR. A. L. SHARP AFWL(WLAA)   SAB 180
C    KIRTLAND AFB, N. MEX. PH. 2471711 EX. 2438.                       SAB 190
```

```
C     **************************************************************
      COMMON /TRI/ PZ,PG,PB,HZ,HG,HB,PZR,PGR,PBR,W,SR,FR,SRE,ALP1ER,ALP1
     1E,R90,PBPZ,R,PBRW,INCOMP
      COMMON /SENSE/ CFS
      COMMON /COLO/ CF11(2),CF22(2),CF33(2),CF44(2),CF55(2),CFR(6),CF1(7
     1),CF2(7),CF3(7)
      COMMON /TAB/ TAB1I(69),TAB1D(69),TAB2I(62),TAB2D(62),TAB3I(69),TAB
     13D(69),TAB4I(18),TAB4D(18),TAB5I(26),TAB5D(26),TAB6I(69),TAB6D(69)
      COMMON /CON/ UL2(7),UL3(8),UL4(8),UL5(5),C4(7),C5(7),C6(7),C7(8),C
     18(8),C9(8),C10(8),C11(8),C12(8),P2(8),P3(8),A1E(41),ACF(13),CF(13)
      COMMON /PNT/ RA,STO,PHIR,SHB,ST,XKK,ALTSRG,HT,XKKX
      COMMON /OVP/ DELP,DELPD,DELPR,CONT,NIT,XITER
      COMMON /TBLKUP/ L1,LF,NA(6),XL(100),NNEX
      DIMENSION P(3), H(3), PR(3), TEMP(3), TM(3), RHO(3), SS(3), SUBS(1
     12)
      DIMENSION IHED1(5,3), IHED2(3,2), ID(28), STP(100), HTP(100)
      DIMENSION ID1(2)
      EQUIVALENCE (P,PZ), (H,HZ), (PR,PZR), (RHOZ,RHO(1)), (RHOG,RHO(2))
     1, (RHOB,RHO(3)), (SSZ,SS(1)), (SSG,SS(2)), (SSB,SS(3))
      DATA (IHED1(J),J=1,5)/50H          OVERPRESSURE SOLUTION
     1       /
      DATA (IHED1(J),J=6,10)/50H          TRIPLE POINT PATH SOLUTION
     1        /
      DATA (IHED1(J),J=11,15)/50H          RANGE SOLUTION
     1         /
      DATA (IHED2(J),J=1,3)/30H                              /
      DATA (IHED2(J),J=4,6)/30H   (WITH TEMPERATURE CHANGE)  /
      DATA (ID(J),J=1,28)/2HSR,2HHZ,2HHG,2HHB,1HW,3HTSA,2HFR,3HSFV,4HDEL
     1P,3HPMV,3HP00,5HSDELP,4HRBAR,1HR,4HALFA,3HSRE,5HALP1E,3HSTO,2HRA,2
     2HST,2HHT,5HPOOOP,4HPDMV,4HPDOD,1HQ,4HRHOZ,3HSSZ,3HCF /
      DATA (ID1(J),J=1,2)/4HHORF,4HHORN/
      DATA CF11/.599829717,.59990265538/
      DATA CF22/-.813121060,-.83454567092/
      DATA CF33/.0537960684,.20153883084/
      DATA CF44/.178602593,-.042235077752/
      DATA CF55/.137932981,.0038268822887/
      DATA CFS/1.85/
      DATA CFR/.5,.8,1.2,1.6,3.0,10./
      DATA CF1/0.,-.6444,-.1625,-.05,-.007,.0001111,0./
      DATA CF2/.6499,1.2943,.4725,.2050,.0424,-.003466,0./
      DATA CF3/.007076,-.154024,.195,.354,.50408,.57704,.591/
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DATA IDOIT/0/
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INTEGER SAVCASE
C     * * * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * *
      SR=SRCM
      HZ=HZCM+HGCM
      HG=HGCM
      HB=HBCM+HGCM
      W=WCM
```

```
      DELP=DELPCM
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      NPROB=0
      R10=10.0/57.296
      R90=90.0/57.296
      R130=130.0/57.296
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      IF (IDOIT.NE.0) GO TO 20
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DO 10 J=1,18
      TAB4I(J)=TAB4I(J)/57.296
10    TAB4O(J)=TAB4O(J)/57.296
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
20    IDOIT=1
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      NN=70
      DO 30 J=1,69
      N=NN-J
      TAB3I(J)=TAB1O(N)
30    TAB3O(J)=TAB1I(N)
40    PMVD=0.
      PMV=0.
      NPROB=NPROB+1
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     READ (5,1190) XIHB,XMHB,XIHZ,XMHZ
C     IF (ENDFILE 5) 1130,40
      IF (NPROB.GE.2) GO TO 1130
      XIHB=XMHB=XIHZ=XMHZ=0.0
C     40 IF(NCASE.GT.0) WRITE(6,1140) NCASE
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      CHK=1.
      PSL=2116.217/144.
50    N=0
      XPRNT=0.
      NIT=0
      XITER=0.
      XKKX=0.
      OVPR=0.
      CONT=0.
      DELPO=0.
      INCOMP=0
      NPT=0
      JTP=0
      LPL=0
      SRE=0.0
      ALP1E=0.0
      STO=0.0
      RA=0.0
      ST=0.0
      HT=0.0
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
      IF (TSA.GT.0.0) TSACM=TSA
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * * * * * *
      TSA=0.0
      IF (CHK-1.) 60,110,40
60    IF (XIHB) 80,80,70
70    HB=HB+XIHB
      SAVBURS=HB
      IF (HB-XMHB) 80,80,40
80    IF (XIHZ) 100,100,90
90    HZ=HZ+XIHZ
      SAVALT=HZ
      IF (HZ-XMHZ) 120,120,40
100   IF (XIHB.EQ.0..AND.XIHZ.EQ.0.) GO TO 40
      GO TO 120
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * * * * * *
110   KCASE=2
C     110 READ (5,1180) KCASE,KTEMP,KDELT,SR,HZ,HG,HB,W,DELP
      KTEMP=KDELT=0
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * * * * * *
      SRSV=SR
      WOR=W
120   IF (HB.LT.25000.) GO TO 130
      CALL SETUP (ACF,1,2,13,0,0,0,0,0)
      CALL MACURE (CF,HB,0,0,0,0,0,LER,CFF)
      W=CFF*WOR
130   IF (CHK.EQ.0.) GO TO 200
      IF (KTEMP.EQ.0) GO TO 150
      IF (XIHB.EQ.0..AND.XIHZ.EQ.0.) GO TO 140
      KTEMP=0
      WRITE (6,1340)
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * * * * * *
140   WRITE (6,1140)
C     140 READ (5,1200) TEMPS,TM
      GO TO 1130
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * * * * * *
150   IF (KDELT.EQ.0) GO TO 170
      IF (XIHB.EQ.0..AND.XIHZ.EQ.0.) GO TO 160
      KDELT=0
      WRITE (6,1350)
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * * * * * *
160   WRITE (6,1150)
C     160 READ (5,1200) SUBS
      GO TO 1130
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * * * * * *
170   IF (KCASE.EQ.3) GO TO 190
      IF (KCASE.EQ.0) KOPT=0
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * * * * * *
C     IF (KCASE.EQ.2) READ (5,1210) KOPT
      KOPT=1
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * * * * * *
      KCASE=KCASE+1
```

```
      IF (KCASE-3) 200,200,180                                        SAB1730
180   PRINT 1170                                                      SAB1740
      GO TO 50                                                        SAB1750
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *SAB1760
190   WRITE (6,1160)                                                  SAB1770
C     190 READ (5,1190) PMV                                           SAB1780
      GO TO 1130                                                      SAB1790
200   K=KTEMP+1                                                       SAB1800
      IF ((XIHB.GT.0..OR.XIHZ.GT.0.).AND.K.GT.1) GO TO 210            SAB1810
      WRITE (6,1200) (IHED1(J,KCASE),J=1,5),(IHED2(J,K),J=1,3)        SAB1820
      IF (KCASE.NE.2) GO TO 210                                       SAB1830
      IF (KDELT.EQ.0) GO TO 210                                       SAB1840
      IF (KTEMP.EQ.0) GO TO 210                                       SAB1850
      WRITE (6,1210)                                                  SAB1860
      KTEMP=0                                                         SAB1870
210   IF (HB.LT.250000.) GO TO 220                                    SAB1880
      WRITE (6,1260)                                                  SAB1890
      WRITE (6,1270) HB                                               SAB1900
      IF (KCASE.EQ.2) GO TO 1060                                      SAB1910
      GO TO 50                                                        SAB1920
220   IF (CONT.GT.0.) GO TO 250                                       SAB1930
      IF (KCASE-1) 230,230,250                                        SAB1940
230   ALTM=ABS(HZ-HB)                                                 SAB1950
      IF (SRSV-ALTM) 240,250,250                                      SAB1960
240   WRITE (6,1380) HB,HZ,SR                                         SAB1970
      IF (XIHB.GT.0..OR.XIHZ.GT.0.) GO TO 50                          SAB1980
      GO TO 40                                                        SAB1990
250   IF (XITER) 260,260,390                                          SAB2000
260   IF (CHK) 270,270,280                                            SAB2010
270   IF (OVPR-1.) 310,340,390                                        SAB2020
280   SR=SR/1000.                                                     SAB2030
      DO 290 J=1,3                                                    SAB2040
      CALL ATMOS (H(J),TEMP(J),DEN,RHO(J),TR,PR(J),SS(J),VC,KER)      SAB2050
      IF (KER.NE.1) GO TO 300                                         SAB2060
290   P(J)=PR(J)*PSL                                                  SAB2070
      GO TO 350                                                       SAB2080
300   WRITE (6,1250) J,KER                                            SAB2090
      GO TO 1060                                                      SAB2100
310   IF (XIHB) 330,330,320                                           SAB2110
320   CALL ATMOS (HB,TEMP(3),DEN,RHOB,TR,PBR,SSB,VC,KER)              SAB2120
      IF (KER.NE.1) GO TO 300                                         SAB2130
      PB=PBR*PSL                                                      SAB2140
330   IF (XIHZ) 350,350,340                                           SAB2150
340   CALL ATMOS (HZ,TEMP(1),DEN,RHOZ,TR,PZR,SSZ,VC,KER)              SAB2160
      IF (KER.NE.1) GO TO 300                                         SAB2170
      PZ=PZR*PSL                                                      SAB2180
350   IF (KTEMP.EQ.0) GO TO 390                                       SAB2190
      PSL=14.696*TEMPS/518.67                                         SAB2200
      IF (OVPR) 360,360,380                                           SAB2210
360   DO 370 J=1,3                                                    SAB2220
      P(J)=P(J)*TEMP(J)/TM(J)                                         SAB2230
```

```
      PR(J)=P(J)/PSL                                                  SAB2240
      RHO(J)=RHO(J)*TM(J)/TEMP(J)                                     SAB2250
370   SS(J)=49.02*TM(J)**.5                                           SAB2260
      GO TO 390                                                       SAB2270
380   SS(1)=49.02*TM(1)**.5                                           SAB2280
390   IF (PMVD.EQ.0.) GO TO 430                                       SAB2290
      IF (XPRNT-1.) 400,430,410                                       SAB2300
400   IF (XITER-1.) 420,410,430                                       SAB2310
410   WRITE (6,1280) PMVD                                             SAB2320
      GO TO 430                                                       SAB2330
420   WRITE (6,1280) PMVD                                             SAB2340
      IF (KOPT.EQ.0) KOPT=1                                           SAB2350
      WCZ1=PMV/SSZ                                                    SAB2360
      WCZ2=WCZ1**2.                                                   SAB2370
      RADIC=SQRT(.36*WCZ2+1.)                                         SAB2380
      DELP=PZ*(.84*WCZ2+1.4*WCZ1*RADIC)                               SAB2390
430   CHK=0.                                                          SAB2400
      OVPR=0.                                                         SAB2410
      PBPZ=PZ/PB                                                      SAB2420
      SSZR=SSZ/1116.4437                                              SAB2430
      PBRW=(PBR/W)**.333333                                           SAB2440
      PZRW=(PZR/W)**.333333                                           SAB2450
440   GO TO (470,450,480), KCASE                                      SAB2460
450   IF (HF.EQ.0..AND.N.GT.0) GO TO 50                               SAB2470
      RF=145.*W**.4                                                   SAB2480
      ALTF=HB-HG                                                      SAB2490
      IF (RF-ALTF) 490,460,460                                        SAB2500
460   WRITE (6,1390) W,HB                                             SAB2510
      GO TO 1060                                                      SAB2520
470   IF (SR.NE.0.0) GO TO 490                                        SAB2530
      PRINT 1180                                                      SAB2540
      GO TO 50                                                        SAB2550
480   IF (W.NE.0.0) GO TO 890                                         SAB2560
      PRINT 1190                                                      SAB2570
      GO TO 50                                                        SAB2580
490   CALL TRIPNT (KCASE)                                             SAB2590
      IF (INCOMP.EQ.1) GO TO 1060                                     SAB2600
      GO TO (550,500), KCASE                                          SAB2610
500   ST0=SRE*SIN(ALP1ER)                                             SAB2620
      ALPHIR=0.0                                                      SAB2630
510   CALL SETUP (TAB4I,1,2,18,0,0,0,0,0)                             SAB2640
      CALL MACURE (TAB4D,ALPHIR,0,0,0,0,0,LER,PHIR)                   SAB2650
      ALPHI=ALPHIR+ALP1ER                                             SAB2660
      RA=SRE*(SIN(R90+ALP1ER-PHIR)/SIN(R90-ALPHI+PHIR))               SAB2670
      RA=ABS(RA)                                                      SAB2680
      JTP=JTP+1                                                       SAB2690
      ST=RA*COS(R90-ALPHI)                                            SAB2700
      STP(JTP)=ST                                                     SAB2710
      IF (NPT.GT.0) GO TO 520                                         SAB2720
      IF (JTP.EQ.1) GO TO 520                                         SAB2730
      IF (STP(JTP).LE.STP(JTP-1)) NPT=JTP-1                           SAB2740
```

```
520   HT=(ST-STO)*(SIN(PHIR)/COS(PHIR))
      HTP(JTP)=HT
      RBAR=RA*PBRW
      CALL SETUP (TAB6I,1,2,69,0,0,0,0,0)
      CALL MACURE (TAB6D,RBAR,0,0,0,0,0,LER,TSACAP)
      TSA=TSACAP/(SSZR*PZRW)
      IF (N.EQ.0) GO TO 530
      IF (ALPHIR.EQ.0.0) WRITE (6,1230) ID(4),HB,ID(5),W,ID(16),SRE,ID(1
     17),ALP1E,ID(18),STO,ID(3),HG
      WRITE (6,1240) ID(19),RA,ID(20),ST,ID(21),HT,ID(6),TSA
      GO TO 540
530   IF (ALPHIR.EQ.0.0) WRITE (6,1220) ID(4),HB,ID(5),W,ID(16),SRE,ID(1
     17),ALP1E,ID(18),STO,ID(3),HG
      WRITE (6,1240) ID(19),RA,ID(20),ST,ID(21),HT,ID(6),TSA
540   ALPHIR=ALPHIR+R10
      IF (R130-ALPHIR) 1020,510,510
550   PBRWFR=(PBR/(W*FR))**.333333
      PZRWFR=(PZR/(W*FR))**.333333
      RBAR=SR*PBRWFR
      IF (KOPT.EQ.1) GO TO 570
      CALL SETUP (TAB1I,1,2,69,0,0,0,0,0)
      CALL MACURE (TAB1D,RBAR,0,0,0,0,0,LER,SDELP)
      IF (HZ.NE.HB) GO TO 560
      DELP=SDELP*PBR
      ALFA=1.0
      GO TO 570
560   CALL SETUP (TAB2I,1,2,62,0,0,0,0,0)
      CALL MACURE (TAB2D,RBAR,0,0,0,0,0,LER,ALFA)
      DELP=SDELP*PBR*PBPZ**ALFA
570   EPSILO=DELP/PZ
      DO 580 J=1,7
      IF (RBAR.LT.UL2(J)) GO TO 590
580   CONTINUE
      TDPZ=.252609+(1.0/11.21)*ALOG(RBAR)
      GO TO 600
590   TDPZ=C4(J)*RBAR**2+C5(J)*RBAR+C6(J)
600   DO 610 J=1,8
      IF (RBAR.LT.UL3(J)) GO TO 620
610   CONTINUE
      J=8
620   TMV=C7(J)*RBAR**P2(J)+C8(J)*RBAR+C9(J)
      DO 630 J=1,8
      IF (RBAR.LT.UL4(J)) GO TO 640
630   CONTINUE
      J=8
640   TDRZ=C10(J)*RBAR**P3(J)+C11(J)*RBAR+C12(J)
      PDOOP=TDPZ/(SSZR*PZRWFR)
      PMV=5.0*EPSILO*SSZ/(7.0*(1.0+6.0*EPSILO/7.0)**.5)
      PDMV=TMV*PDOOP
      POD=RHOZ*(7.0+6.0*EPSILO)/(7.+EPSILO)
      PDOD=TDRZ*PDOOP
```

```
      SFV=SSZ*(1.0+6.0*EPSILO/7.0)**.5
      Q=2.5*DELP**2.0/(7.0*PZ+DELP)
      CALL SETUP (TAB6I,1,2,69,0,0,0,0,0)
      CALL MACURE (TAB6D,RBAR,0,0,0,0,0,LER,TSACAP)
      TSA=TSACAP/(SSZR*PZRWFR)
      IF (KOPT.EQ.1) GO TO 830
      HORF=SQRT((SR*1000.)**2-((ABS(HZ-HB))**2))
      HORN=HORF*.000164
      IF (CONT.EQ.0.) GO TO 730
      NIT=NIT+1
      IF (NIT-1) 730,730,650
650   IF (PMVD) 670,670,660
660   PVPD=ABS(PMV-PMVD)
      IF (PVPD-.01) 730,730,690
670   DPDD=ABS(DELP-DELPD)
      IF (DPDD-.01) 730,730,680
680   DELPR=DELP
      SRS=ABS(HZ-HB)
      SRS=SRS*(DELPR/DELPD)**.33
      GO TO 700
690   SRS=ABS(HZ-HB)
      SRS=SRS*(PMV/PMVD)**.33
700   IF (HB.GT.HZ) GO TO 710
      HZ=HB+SRS
      GO TO 720
710   HZ=HB-SRS
720   OVPR=1.0
      IF (NIT.GT.1) XPRNT=1.
      GO TO 970
730   WRITE (6,1220) IO(1),SR,IO(2),HZ,IO(3),HG,IO(4),HB,IO(5),W
      WRITE (6,1240) IO(6),TSA,IO(7),FR,IO(8),SFV,IO(25),Q
      WRITE (6,1240) IO(9),DELP,IO(10),PMV,IO(11),POD,IO(26),RHOZ
      WRITE (6,1240) IO(22),PDOOP,IO(23),PDMV,IO(24),PDOD,IO(27),SSZ
      WRITE (6,1240) IO(12),SDELP,IO(13),RBAR,IO(14),R,IO(15),ALFA
      WRITE (6,1240) IO1(1),HORF,IO1(2),HORN
      IF (HB.LT.25000.) GO TO 740
      WRITE (6,1240) IO(28),CFF
      WRITE (6,1290) WOR
      GO TO 750
740   WRITE (6,1300)
750   IF (CONT.EQ.0.) GO TO 1060
      IF (NIT-1) 650,650,760
760   IF (PMVD) 770,770,800
770   WRITE (6,1370)
      WRITE (6,1330)
      WRITE (6,1360) NIT
      WRITE (6,1370)
      IF (XIHB.GT.0..OR.XIHZ.GT.0.) GO TO 780
      IF (KOELT.GT.0) GO TO 790
      GO TO 1060
780   KCASE=SAVCASE
```

```
      HZ=SAVALT
      HB=SAVBURS
      DELP=SAVDELP
      GO TO 1060
790   KCASE=SAVCASE
      HZ=SAVALT
      DELP=SAVDELP
      GO TO 1060
800   WRITE (6,1370)
      WRITE (6,1320)
      WRITE (6,1360) NIT
      WRITE (6,1370)
      IF (XIHB.GT.0..OR.XIHZ.GT.0.) GO TO 810
      IF (KDELT.GT.0) GO TO 820
      GO TO 1060
810   KCASE=SAVCASE
      HZ=SAVALT
      HB=SAVBURS
      DELP=SAVDELP
      GO TO 1060
820   KCASE=SAVCASE
      HZ=SAVALT
      DELP=SAVDELP
      GO TO 1060
830   IF (PMVD) 860,860,840
840   PMVR=PMV
      NIT=NIT+1
      XITER=XITER+1.
      IF (ABS(PMVR-PMVD)-.01) 860,860,850
850   DELP=DELP*(PMVD/PMVR)**.33
      IF (NIT.GT.1) GO TO 210
      GO TO 200
860   WRITE (6,1240) ID(6),TSA,ID(8),SFV,ID(25),Q,ID(10),PMV
      WRITE (6,1240) ID(11),POD,ID(26),RHOZ,ID(22),PDDOP,ID(23),PDMV
      WRITE (6,1240) ID(24),PDOD,ID(27),SSZ,ID(14),R,ID(15),ALFA
      IF (HB.LT.25000.) GO TO 870
      WRITE (6,1240) ID(28),CFF
      WRITE (6,1290) WOR
      GO TO 880
870   WRITE (6,1300)
880   IF (XITER.EQ.0.) GO TO 1060
      WRITE (6,1370)
      WRITE (6,1320)
      WRITE (6,1360) NIT
      WRITE (6,1370)
      GO TO 1060
890   ALFA=0.0
      SDELP=DELP/(PBR*PBPZ**ALFA)
      DDELP=-DELP*ALOG(PBPZ)/(PBR*PBPZ**ALFA)
900   J=1
      IF (SDELP-CFS) 920,910,910
```

```
910   J=2
920   PPP=CF11(J)+CF22(J)*ALOG10(SOELP)+CF33(J)*(ALOG10(SOELP)**2)+CF44(
     1J)*(ALOG10(SOELP)**3)+CF55(J)*(ALOG10(SOELP)**4)
      RBAR=10.**PPP
      IF (HB.EQ.HZ) GO TO 960
      AA=ALOG10(2.71828)/SOELP
      BB=2.*ALOG10(SOELP)*AA
      CC=3.*(ALOG10(SOELP)**2)*AA
      DD=4.*(ALOG10(SOELP)**3)*AA
      OPOX=CF22(J)*AA+CF33(J)*BB+CF44(J)*CC+CF55(J)*DD
      ORBAR=RBAR*ALOG(10.)*OPOX
      CF4=0.
      DO 930 II=1,6
      IF (RBAR-CFR(II)) 940,930,930
930   CONTINUE
      II=7
      CF4=-.038
940   SMF=(CF1(II)*RBAR+CF2(II))*RBAR+CF3(II)+CF4*ALOG10(RBAR)
      CAPO=2.*CF1(II)*RBAR+CF2(II)+CF4*ALOG10(2.71828)/RBAR
      CAPQ=CAPO*ORBAR*OOELP
      ALFO=ALFA
      ALFA=(SMF-ALFO*CAPQ)/(1.-CAPQ)
      IF (ABS(ALFA-ALFO)-.001) 960,960,950
950   SOELP=DELP/(PBR*PBPZ**ALFA)
      OOELP=-OELP*ALOG(PBPZ)/(PBR*PBPZ**ALFA)
      GO TO 900
960   SR=RBAR*(W/PBR)**.333333
      CALL TRIPNT (KCASE)
      IF (XKK.LE.1.0) GO TO 980
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      IPROB1=1
      IF (TSA.GT.0.0) TSACM=TSA
      GO TO 1130
970   HABS=ABS(HZ-HB)
      SR=SQRT(HABS**2+(HABS*.00001)**2)
      SR=SR/1000.
      KCASE=1
      KOPT=0
      IF (NIT.GT.1) GO TO 210
      GO TO 200
980   IF (FR.NE.1.0) SR=RBAR*(FR*W/PBR)**.333333
      IF (XKK-1.) 990,1000,1000
990   HORF=SQRT((SR*1000.)**2-((ABS(HZ-HB))**2))
      HORN=HORF*.000164
      GO TO 1010
1000  HORF=0.
      HORN=0.
1010  WRITE (6,1220) ID(5),W,ID(9),OELP,ID(2),HZ,ID(3),HG,ID(4),HB
      WRITE (6,1240) ID(1),SR,ID(13),RBAR,ID(12),SOELP,ID(7),FR
      WRITE (6,1240) ID1(1),HORF,ID1(2),HORN
      IF (KOPT.EQ.1) GO TO 550
```

```
1020  IF (HB.LT.25000.) GO TO 1030                                         SAB4790
      WRITE (6,1240) ID(28),CFF                                           SAB4800
      WRITE (6,1290) WOR                                                  SAB4810
      GO TO 1040                                                          SAB4820
1030  WRITE (6,1300)                                                      SAB4830
1040  IF (JTP.EQ.0.) GO TO 1060                                           SAB4840
      LPL=LPL+100                                                         SAB4850
      WRITE (6,1310) JTP,NPT                                              SAB4860
      DUMY1=AMAX1(STP(NPT),HTP(NPT))-.2                                   SAB4870
      CALL GRAPH (3,1,10,10,9HHOR RANGE,8HALTITUDE,LPL,DUMY1,DUMY1,9.,9.SAB4880
     1,1.,1.)                                                             SAB4890
      CALL GRAPH (5,NPT,10,10,9HHOR RANGE,8HALTITUDE,LPL,STP,HTP,9.,9.,1SAB4900
     1.,1.)                                                               SAB4910
      CALL GRAPH (0)                                                      SAB4920
      DO 1050 I=1,JTP                                                     SAB4930
      STP(JTP)=0.                                                         SAB4940
1050  HTP(JTP)=0.                                                         SAB4950
1060  IF (XIHB.GT.0..OR.XIHZ.GT.0.) GO TO 50                              SAB4960
      CHK=2.                                                              SAB4970
      IF (KDELT.LE.0) GO TO 50                                            SAB4980
      INCOMP=0                                                            SAB4990
      N=N+1                                                               SAB5000
      IF (N.GT.12) GO TO 50                                               SAB5010
      GO TO (1100,1070,1110), KCASE                                       SAB5020
1070  HB=SUBS(N)                                                          SAB5030
      IF (HB.LT.25000..OR.HB.GE.250000.) GO TO 1080                       SAB5040
      CALL SETUP (ACF,1,2,13,0,0,0,0,0)                                   SAB5050
      CALL MACURE (CF,HB,0,0,0,0,0,LER,CFF)                               SAB5060
      WOR=W                                                               SAB5070
      W=CFF*W                                                             SAB5080
      GO TO 1090                                                          SAB5090
1080  IF (HB.LT.250000.) GO TO 1090                                       SAB5100
      WRITE (6,1260)                                                      SAB5110
      WRITE (6,1270) HB                                                   SAB5120
      GO TO 1060                                                          SAB5130
1090  JTP=0                                                               SAB5140
      NPT=0                                                               SAB5150
      CALL ATMOS (HB,TEMP(3),DEN,RHOB,TR,PBR,SSB,VC,KER)                  SAB5160
      IF (KER.NE.1) GO TO 300                                             SAB5170
      PB=PBR*PSL                                                          SAB5180
      NIT=0                                                               SAB5190
      GO TO 430                                                           SAB5200
1100  SR=SUBS(N)/1000.0                                                   SAB5210
      NIT=0                                                               SAB5220
      GO TO 440                                                           SAB5230
1110  IF (HB.LT.25000.) GO TO 1120                                        SAB5240
      WOR=SUBS(N)                                                         SAB5250
      W=SUBS(N)*CFF                                                       SAB5260
      NIT=0                                                               SAB5270
      IF (W.GT.0.) GO TO 430                                              SAB5280
      GO TO 440                                                           SAB5290
```

```
1120  W=SUBS(N)
      NIT=0
      IF (W.GT.0.) GO TO 430
      GO TO 440
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
1130  SRCM=SR
      HZCM=HZ
      HGCM=HG
      HBCM=HB
      WCM=W
      DELPCM=DELP
      TSA=TSACM
      RETURN
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     1220 FORMAT (1H1,45X,5A10,/,55X,3A10,///)
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
1140  FORMAT (///17H  STATEMENT 140  ///)
1150  FORMAT (///17H  STATEMENT 160  ///)
1160  FORMAT (///17H  STATEMENT 190  ///)
1170  FORMAT (1H1,////,10X,65HCASE INPUT ERROR - KCASE GREATER THAN 3 -
     1PROCEEDING TO NEXT CASE)
1180  FORMAT (1H ,////,10X,80HSLANT RANGE FOUND EQUAL TO ZERO IN OVERPRE
     1SSURE SOLUTION PROCEEDING TO NEXT CASE)
1190  FORMAT (1H ,////,10X,67HYIELD FOUND EQUAL TO ZERO IN RANGE SOLUTIO
     1N PROCEEDING TO NEXT CASE)
1200  FORMAT (1H ,45X,5A10,/,55X,3A10,///)
1210  FORMAT (1H0,20X,47HTHIS CASE MUST BE RUN WITH STANDARD TEMPERATURE
     1)
1220  FORMAT (1H0,6(4X,A6,1PE12.5))
1230  FORMAT (1H1,////,6(4X,A6,1PE12.5))
1240  FORMAT (1H0,22X,4(4X,A6,1PE12.5))
1250  FORMAT (1H0,8HATMOS ER,I2,2H =,I3)
1260  FORMAT (1H0,///,48X,37HBLAST EFFECTS ARE ESSENTIALLY REDUCED,//,48
     1X,24HTO ZERO AT THIS ALTITUDE)
1270  FORMAT (1H0,47X,5HHB = ,1PE12.5)
1280  FORMAT (56X,12H(GUST INPUT),//,56X,5HPMV =,E12.5,/)
1290  FORMAT (1H0,26X,36HORIGINAL WEAPON YIELD AS INPUTED(KT),1PE12.5)
1300  FORMAT (1H0,22X,39HYIELD CORRECTION FACTOR IS EQUAL TO ONE)
1310  FORMAT (1H0,6HJTP = ,I6,//,1X,6HNPT = ,I6)
1320  FORMAT (1H0,9X,55HTHE GUST VELOCITY RECEIVED IS WITHIN ONE PERCENT
     1 OF THE,//,10X,49HDESIRED VALUE - NO FURTHER ITERATION IS NECESSAR
     2Y)
1330  FORMAT (1H0,9X,54HTHE OVERPRESSURE RECEIVED IS WITHIN ONE PERCENT
     1OF THE,//,10X,49HDESIRED VALUE - NO FURTHER ITERATION IS NECESSARY
     2)
1340  FORMAT (1H1,5/,1X,134(1H*),3/,10X,51HTHIS PROBLEM MUST BE RUN WITH
     1 A STANDARD ATMOSPHERE,//,10X,46HIF NON STANDARD ATMOSPHERE IS STI
```

```
      2LL DESIRED AN,//,10X,55HINCREMENTAL HEIGHT OF BURST OR RECEIVER MA
      3Y NOT BE USED,3/,10X,40HE.G. SET XIHB AND XIHZ EQUAL TO ZERO AND,/
      4/,10X,31HRUN EACH CONDITION INDIVIDUALLY,3/,10X,53HTHE FOLLOWING O
      5UTPUT IS VALID FOR STANDARD ATMOSPHERE,3/,1X,134(1H*))
 1350 FORMAT (1H1,5/,1X,134(1H*),3/,10X,36HTHE KDELT = 1 OPTION MAY NOT
      1BE USED,//,10X,37HWITH AN INCREMENTAL BURST OR RECEIVER,3/,10X,41H
      2E.G. IF KDELT = 1 - XIHB AND XIHZ MUST BE,//,10X,13HEQUAL TO ZERO,
      33/,10X,33HIN THE FOLLOWING OUTPUT KDELT = 0,3/,1X,134(1H*))
 1360 FORMAT (1H0,9X,23HNUMBER OF ITERATIONS = ,I2)
 1370 FORMAT (1H0,134(1H*))
 1380 FORMAT (10/,45X,40(1H*),3/,45X,40H*    AN OVERPRESSURE SOLUTION CA
      1NNOT    *,/,45X,40H*    BE OBTAINED WITH THE GIVEN        *,/,45X,4
      20H*    INPUT GEOMETRY                        *,//,60X,4HHB =,E12.5,/,6
      30X,4HHZ =,E12.5,/,60X,4HSR =,E12.5,//,45X,40H*    THE PROGRAM WILL
      4 PROCEED WITH      *,/,45X,40H*    THE NEXT CASE
      5  *,3/,45X,40(1H*))
 1390 FORMAT (10/,45X,40(1H*),3/,45X,40H*     NO TRIPLE POINT CALCULATIO
      1N      *,/,45X,40H*     REQUIRED - BURST AND YIELD       *,/,45X,4
      20H*     COMBINATION REQUIRES GROUND       *,/,45X,40H*     BURST CR
      3ITERIA                   *,//,60X,5HW  = ,F10.2,/,60X,5HHB = ,F10.
      42,/,60X,15HFR =         1.60,//,45X,40H*     ALL RECEIVERS FALL WITH
      5IN         *,/,45X,40H*     TRIPLE POINT PATH              *,//,4
      65X,40H*     THE PROGRAM WILL PROCEED WITH     *,/,45X,40H*     THE
      7NEXT CASE                      *,3/,45X,40(1H*))
      END
      SUBROUTINE TRIPNT (KCASE)
C     ***************************************************************
C     SUBROUTINE TRIPNT CALCULATES LIMITING ANGLE FOR REGULAR REFLECTION
C         AND PREDICTS WHETHER RECEIVER IS IN    OUT OF THE FUSED SHOCK
C         REGION
C
C     ROUTINE REQUIREMENTS-
C         NUMEROUS PARAMETERS FORM MAIN ROUTINE THROUGH COMMON
C         CALLS SUBROUTINES SETUP AND MACURE
C
C     CALLING SEQUENCE
C             WHERE-
C                   KCASE=1 FOR OVERPRESSURE SOLUTION
C                         2 FOR TRIPLE POINT PATH SOLUTION
C                         3 FOR RANGE SOLUTION
C     ***************************************************************
C         CALL TRIPNT(KCASE)
      COMMON /TRI/ PZ,PG,PB,HZ,HG,HB,PZR,PGR,PBR,W,SR,FR,SRE,ALP1ER,ALP1
     1E,R90,PBPZ,R,PBRW,INCOMP
      COMMON /SENSE/ CFS
      COMMON /COLO/ CF11(2),CF22(2),CF33(2),CF44(2),CF55(2),CFR(6),CF1(7
     1),CF2(7),CF3(7)
      COMMON /TAB/ TAB1I(69),TAB1O(69),TAB2I(62),TAB2O(62),TAB3I(69),TAB
     13O(69),TAB4I(18),TAB4O(18),TAB5I(26),TAB5O(26),TAB6I(69),TAB6O(69)
      COMMON /CON/ UL2(7),UL3(8),UL4(8),UL5(5),C4(7),C5(7),C6(7),C7(8),C
     18(8),C9(8),C10(8),C11(8),C12(8),P2(8),P3(8),A1E(41),ACF(13),CF(13)
```

```
      COMMON /PNT/ RA,STO,PHIR,SHB,ST,XKK,ALTSRG,HT,XKKX
      COMMON /OVP/ DELP,DELPD,DELPR,CONT,NIT,XITER
      DIMENSION ALPHA(41), IHED1(5,3), ID(28)
      DATA (IHED1(J),J=6,10)/50H          TRIPLE POINT PATH SOLUTION
     1       /
      DATA (ID(J),J=1,28)/2HSR,2HHZ,2HHG,2HHB,1HW,3HTSA,2HFR,3HSFV,4HDEL
     1P,3HPMV,3HPOD,5HSOELP,4HRBAR,1HR,4HALFA,3HSRE,5HALPLE,3HSTO,2HRA,2
     2HST,2HHT,5HPDOOP,4HPDMV,4HPDOD,1HQ,4HRHOZ,3HSSZ,3HCFF/
      PGPB=PG/PB
      ALT=(HB-HG)/1000.0
      R=145.0*W**0.4/1000.0
      SHB=ALT*PBRW
      SHBB=ALT/W**.3333
      IF (SHBB.GT.2.5) GO TO 110
      IF (R.LT.ALT) GO TO 30
      FR=1.6
      IF (SR.NE.0.) GO TO 10
      XK=0.
      PRINT 220
      GO TO 20
10    XK=ABS(HZ-HB)/(SR*1000.0)
20    IF ((ABS(XK-1.)).LE..002) XK=1.
      IF (XK.LE.1.0) GO TO 200
      GO TO 170
30    XI=0.0
      RBAR=ALT*PBRW
      CALL SETUP (TAB2I,1,2,62,0,0,0,0,0)
      CALL MACURE (TAB2D,RBAR,0,0,0,0,0,LER,ALFA)
      DO 100 J=1,41
      ALFA=0.
      XI=XI+0.025
      IF (XI.GT.1.0) XI=1.0
      DELPG=PG/XI-PG
40    SDELP=DELPG/(PBR*PGPB**ALFA)
      DDELP=-DELPG*ALOG(PGPB)/(PBR*PGPB**ALFA)
      K=1
      IF (SOELP-CFS) 60,50,50
50    K=2
60    PPP=CF11(K)+CF22(K)*ALOG10(SDELP)+CF33(K)*(ALOG10(SDELP)**2)+CF44(
     1K)*(ALOG10(SDELP)**3)+CF55(K)*(ALOG10(SDELP)**4)
      RBAR=10.**PPP
      AA=ALOG10(2.71828)/SDELP
      BB=2.*ALOG10(SDELP)*AA
      CC=3.*(ALOG10(SDELP)**2)*AA
      DD=4.*(ALOG10(SDELP)**3)*AA
      DPDX=CF22(K)*AA+CF33(K)*BB+CF44(K)*CC+CF55(K)*DD
      DRBAR=RBAR*ALOG(10.)*DPDX
      CF4=0.
      DO 70 II=1,6
      IF (RBAR-CFR(II)) 80,70,70
70    CONTINUE
```

```
      II=7
      CF4=-.038
80    SMF=(CF1(II)*RBAR+CF2(II))*RBAR+CF3(II)+CF4*ALOG10(RBAR)
      CAPO=2.*CF1(II)*RBAR+CF2(II)+CF4*ALOG10(2.71828)/RBAR
      CAPQ=CAPO*DRBAR*DDELP
      ALFO=ALFA
      ALFA=(SMF-ALFO*CAPQ)/(1.-CAPQ)
      IF (ABS(ALFA-ALFO)-.001) 90,90,40
90    ALTSRG=ALT/(RBAR*(W/PBR)**.333333)
      IF (ALTSRG.GT.1.00) GO TO 100
      ALPHA(J)=ACOS(ALTSRG)*57.296
      IF (ALPHA(J).GT.A1E(J)) GO TO 140
100   CONTINUE
110   IF (KCASE.NE.2) GO TO 120
      INCOMP=1
      WRITE (6,230) (IHED1(J,KCASE),J=1,5)
      WRITE (6,240)
      WRITE (6,250) HB
      RETURN
120   FR=1.0
      IF (XITER.GT.1..OR.NIT.GT.1) GO TO 130
      PRINT 260
130   XK=ABS(HZ-HB)/(SR*1000.0)
      IF ((ABS(XK-1.)).LE..002) XK=1.
      IF (XK.LE.1.0) GO TO 200
      GO TO 170
140   IF (J.NE.1) GO TO 150
      ALP1E=A1E(1)
      GO TO 160
150   ALP1E=((A1E(J)-A1E(J-1))*(ALPHA(J-1)-A1E(J-1))/(A1E(J)-A1E(J-1)-AL
     1PHA(J)+ALPHA(J-1))+A1E(J))
160   ALP1ER=ALP1E/57.296
      SRE=ALT/COS(ALP1ER)
      IF (KCASE.EQ.2) GO TO 200
      ALPHI=R90
      IF (HZ.EQ.HB) GO TO 190
      XK=ABS(HZ-HB)/(SR*1000.0)
      IF ((ABS(XK-1.)).LE..002) XK=1.
      IF (XK.LE.1.0) GO TO 180
170   WRITE (6,270) XK
      WRITE (6,280) ID(5),W,ID(9),DELP
      WRITE (6,280) ID(2),HZ,ID(3),HG
      WRITE (6,280) ID(4),HB,ID(1),SR
      WRITE (6,290)
      DELPO=DELP
      CONT=CONT+1.
      WRITE (6,300)
      XKK=XK
      XKKX=XK
      RETURN
180   XKK=XK
```

```
      XK=ASIN(XK)                                                       TPN1290
      IF (HZ.LT.HB) ALPHI=ALPHI-XK                                      TPN1300
      IF (HZ.GT.HB) ALPHI=ALPHI+XK                                      TPN1310
190   ALPHIR=ALPHI-ALP1ER                                               TPN1320
      CALL SETUP (TAB4I,1,2,18,0,0,0,0,0)                               TPN1330
      CALL MACURE (TAB4D,ALPHIR,0,0,0,0,0,LER,PHIR)                     TPN1340
      RA=SRE*COS(ALP1ER-PHIR)/COS(ALPHI-PHIR)                           TPN1350
      RA=ABS(RA)                                                        TPN1360
      FR=1.0                                                            TPN1370
      IF (XKKX.GT.0.) GO TO 210                                         TPN1380
      IF (RA.GT.SR) GO TO 210                                           TPN1390
      CALL SETUP (TAB5I,1,2,26,0,0,0,0,0)                               TPN1400
      CALL MACURE (TAB5D,SHB,0,0,0,0,0,LER,FR)                          TPN1410
      IF (SHB.LT.1.54) GO TO 210                                        TPN1420
      IF ((SR-RA).LT.0.1) GO TO 210                                     TPN1430
      FR=2.33-0.025*RBAR                                                TPN1440
200   XKK=XK                                                            TPN1450
210   RETURN                                                            TPN1460
C                                                                       TPN1470
C                                                                       TPN1480
C                                                                       TPN1490
220   FORMAT (29H SR=0.0 IN SUBROUTINE TRIPNT )                         TPN1500
230   FORMAT (1H1,45X,5A10,///)                                         TPN1510
240   FORMAT (1H0,///,48X,35HINPUT PARAMETERS ARE NOT COMPATIBLE,//,48X,TPN1520
     134HFOR THE TRIPLE POINT PATH SOLUTION)                            TPN1530
250   FORMAT (1H0,47X,5HHB = ,1PE12.5)                                  TPN1540
260   FORMAT (3/,1X,134(1H*),3/,10X,42HSCALED HEIGHT OF BURST IS GREATERTPN1550
     1 THAN 2.5,//,10X,51HGROUND EFFECT AMPLIFICATION FACTOR SET EQUAL TTPN1560
     20 1.0,3/,1X,134(1H*),3/)                                          TPN1570
270   FORMAT (38H ***ARG OF ASIN (X) OUT OF RANGE.    X=,E16.8,////)    TPN1580
280   FORMAT (1H0,22X,2(4X,A6,1PE12.5))                                 TPN1590
290   FORMAT (1X,5/)                                                    TPN1600
300   FORMAT (10X,66HTHE INPUT GIVEN IS INCOMPATABLE WITH A POSSIBLE PHYTPN1610
     1SICAL CONDITION,//,10X,39HTWO ALTERNATE SETS OF OUTPUT ARE GIVEN-,TPN1620
     2/,14X,61H1-RECEIVER DIRECTLY ABOVE OR BELOW THE BURST DEPENDING ONTPN1630
     3 THE,/,16X,57HINITIAL ORIENTATION OF RECEIVER WITH RESPECT TO THE TPN1640
     4BURST,//,14X,64H2-THE ALTITUDE AT WHICH THE DESIRED GUST OR OVERPRTPN1650
     5ESSURE OCCURRS,//,34X,1H*,/,34X,1H*,/,34X,1H*,/,34X,1H*,/,31X,7H**TPN1660
     6*****,/,32X,5H*****,/,33X,3H***,/,34X,1H*)                        TPN1670
      END                                                               TPN1680-
      BLOCK DATA                                                        BD2   10
C     ******************************************************************BD2   20
C     BLOCK DATA CONTAINS TABULATED VALUES USED IN THE MAIN PROGRAM     BD2   30
C     AND IN SUBROUTINE TRIPNT                                          BD2   40
C     ******************************************************************BD2   50
      COMMON /TRI/ PZ,PG,PB,HZ,HG,HB,PZR,PGR,PBR,W,SR,FR,SRE,ALP1ER,ALP1BD2   60
     1E,R90,PBPZ,R,PERW                                                 BD2   70
      COMMON /TAB/ TAB1I(69),TAB1D(69),TAB2I(62),TAB2D(62),TAB3I(69),TABBD2   80
     13D(69),TAB4I(18),TAB4D(18),TAB5I(26),TAB5D(26),TAB6I(69),TAB6D(69)BD2   90
      COMMON /CON/ UL2(7),UL3(8),UL4(8),UL5(5),C4(7),C5(7),C6(7),C7(8),CBD2  100
     18(8),C9(8),C10(8),C11(8),C12(8),P2(8),P3(8),A1E(41),ACF(13),CF(13)BD2  110
```

```
      DATA (TAB1I(J),J=1,69)/.050,.0625,.075,.0875,.100,.125,.150,.175,.BD2 120
     1200,.225,.250,.275,.300,.325,.350,.375,.400,.450,.500,.550,.600,.6BD2 130
     250,.700,.750,.800,.850,.900,.950,1.000,1.105,1.221,1.350,1.492,1.6BD2 140
     349,1.822,2.014,2.226,2.460,2.718,3.004,3.320,3.669,4.055,4.482,4.9BD2 150
     453,5.474,6.050,6.686,7.389,8.166,9.025,9.974,11.023,12.182,13.464,BD2 160
     514.880,16.445,18.174,20.086,22.198,24.532,27.113,29.964,33.115,36.BD2 170
     6598,40.447,44.701,50.000,100.00/                                  BD2 180
      DATA (TAB1D(J),J=1,69)/17200.,8000.,4240.,2490.,1660.,900.,545.,36BD2 190
     10.,246.,186.,144.,107.,88.8,73.5,62.7,54.5,46.0,36.4,29.3,24.1,20.BD2 200
     22,17.2,14.8,12.9,11.3,9.9,8.8,7.9,7.0,6.2,5.2,4.45,3.83,3.30,2.85,BD2 210
     32.48,2.10,1.85,1.62,1.43,1.27,1.12,.98,.87,.77,.68,.60,.53,.47,.41BD2 220
     4,.36,.32,.28,.252,.222,.198,.174,.156,.139,.124,.113,.100,.091,.08BD2 230
     53,.0755,.069,.063,.0574,.0333/                                    BD2 240
      DATA TAB2I/0.,.1,.2,.30,.35,.40,.45,.50,.55,.60,.65,.70,.75,.80,.8BD2 250
     15,.90,.95,1.0,1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5,7.0,7.5,BD2 260
     28.0,8.5,9.0,9.5,10.,15.,20.,25.,30.,35.,40.,45.,50.,55.,60.,65.,70BD2 270
     3.,75.,80.,85.,90.,95.,100.,150.,200.,250.,300.,350.,400.,450.,500.BD2 280
     4/                                                                 BD2 290
      DATA TAB2D/.020,.0625,.125,.1800,.2150,.2460,.2930,.3320,.3650,.39BD2 300
     110,.4150,.4350,.4530,.4690,.4800,.4900,.4990,.5060,.5490,.5620,.56BD2 310
     260,.5670,.5660,.5650,.5630,.5620,.5610,.5605,.5590,.5580,.5575,.55BD2 320
     370,.5560,.5540,.5535,.5530,.5475,.5420,.5390,.5360,.5330,.5310,.52BD2 330
     490,.5275,.5260,.5230,.5220,.5210,.5200,.5190,.5180,.5170,.5160,.51BD2 340
     550,.5080,.5020,.4980,.4940,.49[illegible]890,.4870,.4840/                 BD2 350
      DATA (TAB4I(J),J=1,18),(TAB4D(J),J=1,18)/0.0,5.0,10.0,15.0,20.0,25BD2 360
     1.0,30.0,35.0,40.0,45.0,50.0,60.0,70.0,80.0,90.0,100.0,110.0,120.0,BD2 370
     20.0,0.0,0.25,0.50,1.0,2.0,3.5,5.0,7.0,9.5,12.5,20.0,28.0,36.5,45.3BD2 380
     3,54.5,63.3,72.0/                                                  BD2 390
      DATA (TAB5I(J),J=1,26),(TAB5D(J),J=1,26)/0.0,0.1,0.2,0.3,0.4,0.5,0BD2 400
     1.6,0.7,0.8,0.9,1.0,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2.0,2.1,2.2BD2 410
     2,2.3,2.4,2.5,1.60,1.62,1.75,2.02,2.39,2.90,3.53,4.30,5.00,5.46,5.5BD2 420
     36,5.73,4.45,3.51,2.88,2.45,2.12,1.86,1.65,1.50,1.36,1.25,1.16,1.08BD2 430
     4,1.03,1.00/                                                       BD2 440
      DATA (TAB6I(J),J=1,69)/.050,.0625,.075,.0875,.100,.125,.150,.175,.BD2 450
     1200,.225,.250,.275,.300,.325,.350,.375,.400,.450,.500,.550,.600,.6BD2 460
     250,.700,.750,.800,.850,.900,.950,1.000,1.105,1.221,1.350,1.492,1.6BD2 470
     349,1.822,2.014,2.226,2.460,2.718,3.004,3.320,3.669,4.055,4.482,4.9BD2 480
     453,5.474,6.050,6.686,7.389,8.166,9.025,9.974,11.023,12.182,13.464,BD2 490
     514.880,16.445,18.174,20.086,22.198,24.532,27.113,29.964,33.115,36.BD2 500
     6598,40.447,44.701,50.000,100.00/                                  BD2 510
      DATA (TAB6D(J),J=1,69)/.00062,.00106,.00170,.00255,.00363,.00638,.BD2 520
     100994,.01432,.01954,.02555,.03235,.04032,.04963,.05844,.06853,.079BD2 530
     224,.09055,.11481,.14105,.16905,.19860,.22955,.26173,.29501,.32929,BD2 540
     3.36445,.40041,.43710,.47444,.555,.646,.749,.864,.994,1.14,1.30,1.4BD2 550
     48,1.68,1.90,2.15,2.43,2.73,3.07,3.45,3.86,4.32,4.83,5.40,6.02,6.71BD2 560
     5,7.47,8.32,9.25,10.29,11.43,12.69,14.09,15.64,17.34,19.23,21.32,23BD2 570
     6.62,26.17,28.99,32.11,35.55,39.36,44.10,90.00/                    BD2 580
      DATA (UL2(J),J=1,7)/0.3,.35,.45,0.6,1.7,3.8,10./,(UL3(J),J=1,8)/.3BD2 590
     135,.5,1.0,2.5,7.0,10.0,17.0,0.0/,(UL4(J),J=1,8)/.28,0.5,.85,1.12,1BD2 600
     2.5,4.2,10.0,0.0/,(UL5(J),J=1,5)/.265,.43,1.68,5.867375,0.0/       BD2 610
      DATA (C4(J),J=1,7)/-.724616,-.08,-.02,.416667,-.071254,-.007659,-.BD2 620
```

```
      1001097/
       DATA (C5(J),J=1,7),(C6(J),J=1,7),(C7(J),J=1,8),(C8(J),J=1,8)/.0804
      198,-.110,.137,-.220833,.277330,.380934,.029527,.167056,.1662,.0724
      2,.1450,.021749,.171847,.272444,.160875,-.606061,1.44,.097778,.0085
      392,0.0,-.0000083,76.202,1.613868,-3.554546,-3.16,-.542223,-.114293
      4,-0.00167,0.0,0.0/
       DATA (C9(J),J=1,8),(C10(J),J=1,8),(C11(J),J=1,8),(C12(J),J=1,8)/2.
      10443,3.928788,3.22,1.944445,1.432333,1.064659,1.056289,1.0,1.15384
      26,-6.2972,-1.57143,-3.05556,.618421,.0050667,-.00108276,-3.,-.0038
      346,7.98454,3.03571,6.01944,-1.80976,-.0654667,.0224069,0.0,.014651
      44,-1.637971,-.345,-1.80889,2.35118,1.1148,.854207,1.0/
       DATA (P2(J),J=1,8),(P3(J),J=1,8)/5*2.,1.,3.,-3.,7*2.,-2./
       DATA (A1E(J),J=1,41)/40.0,39.6,39.4,39.2,39.0,39.0,39.0,39.0,39.1,
      139.3,39.5,39.6,39.9,40.0,40.4,40.7,41.0,41.3,41.6,42.0,42.4,42.7,4
      23.0,43.5,44.0,44.5,45.3,46.0,47.0,48.0,49.5,50.8,52.3,54.0,56.0,58
      3.0,60.5,63.5,67.5,74.0,90.0/
       DATA (CF(J),J=1,13)/1.0,.98,.96,.917,.83,.66,.47,.343,.265,.211,.1
      175,.143,0./
       DATA (ACF(J),J=1,13)/25000.,50000.,62500.,75000.,87500.,100000.,11
      12500.,125000.,137500.,150000.,162500.,175000.,250000./
       END
       SUBROUTINE ATMOS (Z,TM,SIGMA,RHO,THETA,DELTA,CA,AMU,K)
C      ****************************************************************
C      CALLING SEQUENCE
C
C      CALL ATMOS(Z,TM,SIGMA,RHO,THETA,DELTA,CA,AMU,K)
C
C      Z     = GEOMETRIC ALTITUDE (FT)
C      TM    = MOLECULAR SCALE TEMPERATURE (DEGREES RANKIN)
C      SIGMA = RATIO OF DENSITY TO THAT AT SEA LEVEL
C      RHO   = DENSITY LB-SEC**2-FT**(-4) OR SLUGS-FT**3
C      THETA = RATIO OF TEMPERATURE TO THAT AT SEA LEVEL
C      DELTA = RATIO OF PRESSURE TO THAT AT SEA LEVEL
C      CA    = SPEED OF SOUND (FT/SEC)
C      AMU   = VISCOSITY COEFFICIENT (LB-SEC-FT**2)
C
C      K = 1 NORMAL,
C        = 2 ALTITUDE GREATER THAN 300000. FT.,
C        = 3 ALTITUDE NEGATIVE,
C        = 4 FLOATING POINT OVERFLOW,
C        = 5 ALTITUDE GREATER THAN 300000. FT. AND FLOATING POINT OVERFL.
C      ****************************************************************
       DIMENSION HPRIMB(11), TMB(11), SIGMAB(11), ALM(11)
       DATA (HPRIMB(I),TMB(I),SIGMAB(I),ALM(I),I=1,11)/0.,518.688,1.00000
      100E00,-0.00356616,36089.239,389.988,2.9706958E-01,0.,82020.997,389
      2.988,3.2665751E-02,0.00164592,154199.480,508.788,1.2117870E-03,0.,
      3173884.510,508.788,5.8677311E-04,-0.00246888,259186.350,298.188,5.
      48677311E-04,0.,295275.590,298.188,1.7928595E-06,0.00219456,344488.
      5190,406.188,9.3921519E-08,0.01097280,524934.380,2386.188,7.7658593
      6E-10,0.00548640,557742.780,2566.188,5.6324877E-10,0.0027432,65616
      77.980,2836.188,2.5726771E-10,0.00192024/
```

```
      DATA Q/0.018744176/,RE/2.0855531E07/,S/198.72/,PZ/2116.2/,AMUZ/3.7
     1372998E-07/,RHOZ/0.0023769/,TMZ/518.688/
      K=1
      IF (Z) 10,30,20
10    K=3
      GO TO 110
20    IF (Z.GT.300000.) K=K+1
30    HPRIM=(RE/(RE+Z))*Z
      DO 40 M=1,11
      IF (HPRIM-HPRIMB(M)) 50,60,40
40    CONTINUE
      M=12
50    M=M-1
60    IF (ALM(M)) 70,80,70
70    TM=TMB(M)+ALM(M)*(HPRIM-HPRIMB(M))
      SIGMA=EXP((1.0+(Q/ALM(M)))*(ALOG(TMB(M)/TM)))*SIGMAB(M)
      GO TO 90
80    TM=TMB(M)
      SIGMA=SIGMAB(M)*EXP(-(Q*(HPRIM-HPRIMB(M)))/TMB(M))
90    RHO=RHOZ*SIGMA
      THETA=TM/TMZ
      DELTA=SIGMA*THETA
      CA=49.02177*SQRT(TM)
      AMU=AMUZ*SQRT(THETA**3)*((TMZ+S)/(TM+S))
      CALL OVERFL (J)
      GO TO (100,110), J
100   K=K+3
110   RETURN
      END
      SUBROUTINE SETUP (X,NEXTR,ND,NA1,NA2,NA3,NA4,NA5,NA6)
C     ******************************************************************
C
C     SUBROUTINE SETUP SETS UP ARRAYS FOR TABLE LOOK UP
C
C     CALLING SEQUENCE-
C
C     CALL SETUP(X,NEXTR,ND,NA1,NA2,NA3,NA4,NA5,NA6)
C       WHERE
C                 X     =  TABLE OF INDEPENDENT VARIABLES
C                 NEXTR=  0  NO EXTRAPOLATION
C                       =  1  EXTRAPOLATION IS DESIRED
C                 ND    =  NUMBER OF DIMENSIONS (WHEN Z=F(X,Y), ND=3)
C                 NA1   =  NO. OF VALUES FOR FIRST  INDEPENDENT VARIABLE
C                 NA2   =  NO. OF VALUES FOR SECOND INDEPENDENT VARIABLE
C                 NA3   =  NO. OF VALUES FOR THIRD  INDEPENDENT VARIABLE
C                 NA4   =  NO. OF VALUES FOR FOURTH INDEPENDENT VARIABLE
C                 NA5   =  NO. OF VALUES FOR FIFTH  INDEPENDENT VARIABLE
C                 NA6   =  NO. OF VALUES FOR SIXTH  INDEPENDENT VARIABLE
C
C     ******************************************************************
      COMMON /TBLKUP/ L1,LF,NA(6),XL(100),NNEX
      DIMENSION X(1), XA(6), NS(5), WJ(32), RATIO(5), NGROUP(5), ITOT(5)
```

```
      DO 10 I=1,NA1
10    XL(I)=X(I)
      NNEX=NEXTR
      NA(1)=NA1
      NA(2)=NA2
      NA(3)=NA3
      NA(4)=NA4
      NA(5)=NA5
      NA(6)=NA6
      L1=2
      LF=ND-1
      RETURN
      END
      SUBROUTINE MACURE (Z,XA1,XA2,XA3,XA4,XA5,XA6,IE,ZR)
C     *****************************************************************
C     SUBROUTINE MACURE EXECUTES AN N DIMENSIONAL TABLE LOOK UP
C     WITH EXTRAPOLATION IF DESIRED
C
C     CALLING SEQUENCE-
C
C     CALL MACURE(Z,XA1,XA2,XA3,XA4,XA5,XA6,IE,ZR)
C       WHERE
C               IE    =   ERROR CODE
C                         0  INTERPOLATION SUCCESSFUL
C                         1  INDEPENDENT VARIABLES NOT IN ASCENDING ORDER
C                         2  FOR I=0, ARGUMENT EXCEEDS LIMITS OF TABLE
C               Z( 1)= F(X1,Y1,Z1)            Z(13)= F(X3,Y1,Z1)
C               Z( 2)= F(X1,Y1,Z2)            Z(14)= F(X3,Y1,Z2)
C               Z( 3)= F(X1,Y2,Z1)            Z(15)= F(X3,Y2,Z1)
C               Z( 4)= F(X1,Y2,Z2)            Z(16)= F(X3,Y2,Z2)
C               Z( 5)= F(X1,Y3,Z1)            Z(17)= F(X3,Y3,Z1)
C               Z( 6)= F(X1,Y3,Z2)            Z(18)= F(X3,Y3,Z2)
C               Z( 7)= F(X2,Y1,Z1)            Z(19)= F(X4,Y1,Z1)
C               Z( 8)= F(X2,Y1,Z2)            Z(20)= F(X4,Y1,Z2)
C               Z( 9)= F(X2,Y2,Z1)            Z(21)= F(X4,Y2,Z1)
C               Z(10)= F(X2,Y2,Z2)            Z(22)= F(X4,Y2,Z2)
C               Z(11)= F(X2,Y3,Z1)            Z(23)= F(X4,Y3,Z1)
C               Z(12)= F(X2,Y3,Z2)            Z(24)= F(X4,Y3,Z2)
C
C     *****************************************************************
      COMMON/TBLKUP/L1,LF,NA(6),X(100),NEXTR
      DIMENSION Z(1), XA(6), NS(5), WJ(32), RATIO(5), NGROUP(5), ITOT(5)
      IE=0
      XA(1)=XA1
      XA(2)=XA2
      XA(3)=XA3
      XA(4)=XA4
      XA(5)=XA5
      XA(6)=XA6
      DO 100 I=1,LF
      L2=L1+NA(I)-2
```

```
      FOUND=0.
      DO 50 J=L1,L2
      IF (X(J).GT.X(J-1)) GO TO 10
      IE=2
      RETURN
10    IF (FOUND.NE.0.) GO TO 50
      IF (XA(I)-X(J-1)) 20,50,50
20    IF (J.GT.L1) GO TO 40
      IF (NEXTR.EQ.0) GO TO 30
      FOUND=1.
      NS(I)=L1-1
      GO TO 50
30    IE=-1
      RETURN
40    FOUND=1.
      NS(I)=J-2
50    CONTINUE
      IF (FOUND) 90,60,90
60    IF (XA(I)-X(L2)) 80,80,70
70    IF (NEXTR.NE.0) GO TO 80
      IE=1
      RETURN
80    NS(I)=L2-1
90    L1=L2+2
100   CONTINUE
C     IN NS(I) IS THE SUBSCRIPT IN THE ARRAY X SUCH THAT
C     X(NS(I)) IS LESS THAN THE ITH ARGUMENT
      DO 110 I=1,LF
      K=NS(I)
      RATIO(I)=(XA(I)-X(K))/(X(K+1)-X(K))
C     IN RATIO(1) IS THE RATIO OF X ARG, RATIO(2)=RATIO OF Y ETC.
110   CONTINUE
      NGROUP(1)=NS(1)
      NSUM=NA(1)
      DO 120 I=2,LF
      NGROUP(I)=NS(I)-NSUM
      NSUM=NSUM+NA(I)
120   CONTINUE
C     IN NGROUP(I) IS THE SUBSCRIPT OF THE ITH VARIABLE SUCH
C     THAT THE TABLE VALUE IS LESS THAN THE CORRESPONDING ARGUMENT
C     THIS IS IN TERMS OF THIS VARIABLE ONLY
C     FOR A FUNCTION OF DEGREE ND WE NEED2**(ND-1) VALUES
C     FROM THE Z ARRAY
      ITOT(LF)=1
      DO 130 I=2,LF
      J=LF-I+1
      ITOT(J)=ITOT(J+1)*NA(J+1)
130   CONTINUE
C     IN ITOT(J) IS THE NUMBER OF LOCATIONS IN THE Z ARRAY NEEDED TO CHA
C     THE JTH SUBSCRIPT
      KF=2**LF
```

```
      MW=-2
      DO 170 I=1,KF,2
      IFIRST=1
      MW=MW+2
      DO 160 J=1,LF
      MM=2**(J-1)
      IF (MOD(MW/MM,2).EQ.0) GO TO 140
      IMON=NGROUP(J)+1
      GO TO 150
140   IMON=NGROUP(J)
150   IFIRST=IFIRST+(IMON-1)*ITOT(J)
160   CONTINUE
      ISEC=IFIRST+ITOT(1)
      WJ(I)=Z(IFIRST)
      WJ(I+1)=Z(ISEC)
170   CONTINUE
      DO 180 I=1,LF
      KF=KF/2
      DO 180 J=1,KF
180   WJ(J)=WJ(2*J-1)+(WJ(2*J)-WJ(2*J-1))*RATIO(I)
      ZR=WJ(1)
      RETURN
      END
      SUBROUTINE GRAPH (IA,JM,IU,JC,IB,IL,IV,BR,AA,AP,BC,BS,AB)
C
C     THIS IS A DUMMY
C
      DIMENSION BR(1), AA(1)
      WRITE (6,10)
      RETURN
C
10    FORMAT (1H ,29H SUBROUTINE GRAPH WAS ENTERED)
      END
```

APPENDIX II

SAMPLE INPUT DECK

This appendix illustrates a sample input deck in figure 2. The reader may compare the data with the descriptions in tables 1 through 16. Output generated by QUANTO for this input appears in appendix III of AFWL-TR-73-242, "QUANTO--A Code to Optimize Weapon Allocations." The remarks which follow will merely point out several of the salient features of the input deck. Parenthetical line numbers on the left of each data card are used for reference.

The first two numbers on the first input data card indicate, respectively, that there are four airbases or targets and three candidate positions at which submarines may be located. The locations of the bases and their centroids and their numbers of runways and aircraft appear on cards 2, 4, 6, and 8. Strings of ones indicate the take-off sequences, since only one type of aircraft is included in this problem. Cards 12 through 14 give the candidate submarine positions, the number of submarines initially located at each point, the number of missiles per submarine, and the type of missile (or submarine) which is (or is permitted) at each point. By adding the numbers of salvos possible from all possible submarine locations, the user may compute the number of weapon groups as six in this example. Adjusting the array dimensions downward to accommodate only four targets and six weapon groups could result in an enormous core reduction.

Program options specified on the first data card include the optimizations of both the submarine positions and the aircraft beddown. The extended form of the output has been requested for illustration purposes only, and the user should never have occasion to request this voluminous output. The final number input on the first data card is the MODE option, which in this case requests that all QUANTO computations be attempted throughout all the optimizations, but if the user-specified time limit for this job is within 30 seconds, then all computational results are to be saved on an output magnetic tape for later use in continuing the job, if desired.

```
                                                                    Line No.
   4    3    1    2    1    2    2    2    1    3                     (1)
  38.25     103.25        0.0        2         15.0
1111111111111111
  48.25      97.50        5.5        1         15.0
111111111111111
  42.85      91.40        0.0        2         15.0
111111111111111
  35.7       85.90        5.5        1          5.0
11111
   1.0        4.5         1.5       60                                  (10)
    .1667      .0833
  36.75      74.0         1    2     1
  28.05      93.45        2    2     1
  45.65     126.05        2    2     1
    .25        .9          .95        .95     300.0     2000.0     1500.0
  14          4.3750    310.0      5.3190     520.0        6.430    810.0
   7.5417  1130.0         8.6528 1475.0         9.7639 1765.0
  10.8750  2050.0        11.9861 2310.0        13.0972 2530.0
  14.2083  2740.0        15.3195 2930.0        16.4305 3110.0
  17.5417  3260.0        17.9861 3290.0                                 (20)
 5000.0      2500.0
  20        PHANTOM PROBLEM PROFILE
    0.0                  0.0           0.0
 2000.0                 20.0           9.0             .1406
 3900.0                 30.0          30.0             .1406
 6200.0                 40.0          92.0             .1946
 9500.0                 50.0         360.0             .2538
13400.0                 60.0         500.0             .3154
15600.0                 66.0         500.01            .3549
17300.0                 68.0         500.02            .3672
 7900.0                 69.0         500.03            .3734
 3200.0                 71.0         500.04            .3863
22750.0                 78.0         500.05            .4355
26800.0                 86.0         500.06            .4620
33000.0                 94.0         500.07            .5636
37000.0             101.0            500.08            .5955
44000.0             109.0            500.09            .6379
49750.0             117.0            500.10            .6617
55900.0             125.0            500.11            .6865
61000.0             132.0           1425.0             .7037
68000.0             140.0           3950.0             .7275
75500.0             148.0           5000.0             .7312     6.2
 7600.0                  .849         60.0            10.0
    0.0       1.0        0.3       10.         5.0     10000.0

                                                                        (46)
    .01       20   100      .0001
   2    2
   3    2
   1    4
   2    2
   3    3                                                               (52)
0000000000000000000000                                                   ↑
                                                                    Line No.'s
```

Figure 2. Sample Input Deck

The data on cards 10 and 11 is self-explanatory. If there were more than one type of aircraft, there would be a card of the form of card 10 for each type of aircraft. Furthermore, since there are dual runway bases, two types of aircraft would cause the need to input eight take-off intervals, since there would then be four ordered pairs of aircraft-type numbers for single runway bases and four for dual runway bases. The eight take-off intervals would require two input cards, since each value occupies 10 columns but only 70 columns are read from each card.

Data cards 15 through 20 indicate data describing the SLBM and its flight time. Fourteen time/range pairs for the missile appear on cards 16 through 20 preceded by the pair count. The user has indicated a maximum range of 2000 NM in card 15 even though the time/range pairs extend up to a range of 3290 NM. In QUANTO, flight time for an individual missile will be computed for all targets falling within the range limits on card 15, using cubic interpolation and extrapolation, if necessary, on the time/range pairs, and targets outside the range limits will not be attacked by that missile. If there were more than one missile type, a set of cards like cards 15 through 20 would have to be input for each type missile.

The aircraft profile and nuclear effects parameters are input on cards 21 through 46. The user has labeled the profile in an unused field in the "count card," card 22, which also indicates that 20 range/time/altitude/velocity/acceleration sets are to follow. In these cards, the user has adjusted the altitude values to be strictly increasing, although this was unnecessary. Also, since card 21 indicates a level-off altitude of 5000 feet, the profile data contains an acceleration value only on the card having 5000 feet altitude. The aircraft will be accelerated from Mach 0.7312 to Mach 0.849 after leveling off according to the parameters on cards 42 and 43. The nuclear effects parameters appear on card 44 followed by two blank cards indicating that the lethal radii and the time of shock arrival are not known from previous runs of similar problems. If there were two types of aircraft for this problem, a set of cards similar to the set of cards 21 through 46 would have to be input for each aircraft type. If there were more than one missile type, a pair of cards like cards 45 and 46 would have to be input for each missile type, even if the lethal radii are unknown.

The convergence parameters and the initial missile allocation which starts the iterative procedure are input on cards 47 through 52. In the initial allocation, the two numbers on a single card correspond to the two salvos from a single submarine, and the five submarines are in the order indicated by cards 12 through 14. The initial laydown, for example, has the second submarine at the second submarine location firing its two salvos at targets one and four in the order of launching.

While this example does not exhaustively illustrate all input options, the user can use figure 2 as a guide to constructing the input deck.